

**MISSION OPERATIONS AND DATA SYSTEMS DIRECTORATE**

**Landsat 7 Processing System (LPS)  
Software Requirements Specification (SRS)**

**Revision 2**

**June 1997**



National Aeronautics and  
Space Administration

Goddard Space Flight Center  
Greenbelt, Maryland

# Landsat 7 Processing System (LPS) Software Requirements Specification (SRS)

## Revision 2

## June 1997

Prepared by:

Approved by:

Jeff Hosler  
Software Manager  
Landsat 7 Processing System  
Code 563.2  
Goddard Space Flight Center

J. B. Martin  
Ground System Manager  
Landsat 7 System  
Code 502  
Goddard Space Flight Center

Concurred by:

Joy Henegar  
Project Manager  
Landsat 7 Processing System  
Code 563.1  
Goddard Space Flight Center

R. Obenschain  
Project Manager  
Landsat 7 System  
Code 430  
Goddard Space Flight Center

Dennis M. Giblin  
Chairman, Code 560 CCB  
Goddard Space Flight Center

---

## Abstract

The Landsat 7 Processing System (LPS) provides Landsat 7 data receipt and processing support to the Landsat 7 program, in conjunction with the Earth Science Mission Operations Project. The LPS receives raw wideband data from the Landsat Ground Station, located at the EROS Data Center (EDC); processes it into level-zero reformatted, browse, and metadata files; and provides them to the Landsat Processes Distributed Active Archive Center, also located at the EDC. The software requirements presented in this document are based on the information contained in the LPS Functional and Performance Specification, LPS System Design Specification, and LPS Operations Concept document.

**Keywords:**

- Landsat 7
- Landsat 7 Processing System (LPS)
- Landsat Ground Station (LGS)
- Landsat Processes Distributed Active Archive Center (LP DAAC)
- functional and performance specification (F&PS)
- Mission Operations and Data Systems Directorate (MO&DSD)



---

## Preface

This document contains the software requirements for the Landsat 7 Processing System (LPS). These requirements are based on an analysis of the requirements contained in the LPS Functional and Performance Specification, LPS System Design Specification, and LPS Operations Concept document. This LPS Software Requirements Specification (SRS), once baselined at or after the system design review and software requirements review, will be controlled by the Mission Operations and Systems Development Division Configuration Control Board and maintained and updated, as required, by the LPS Project.

This SRS was prepared by

Landsat 7 Processing System Project  
Mission Operations and Systems Development Division  
Goddard Space Flight Center  
Greenbelt, MD 20771

<b>GSFC</b>	<b>CNMOS</b>
J. Henegar	D. Crehan
J. Hosler	E. Criscuolo
E. Lee	D. Ferry
R. McIntosh	S. Gottsagen
R. Schweiss	L. Gupta
	A. Herbert
	M. Huang
	H. Kim
	C. Liu
	R. Mazzola
	Y. O
	N. Ottenstein
	M. Reid
	R. Tingley
	V. Trinh
	H. Vinluan
	M. Wallace
	I. Wu



---

## Contents

### Section 1 – Introduction

1.1	Scope .....	1-1
1.2	Applicable Documents .....	1-1
1.2.1	Specification Documents .....	1-1
1.2.2	Reference Documents.....	1-2
1.3	Definitions .....	1-3

### Section 2 – Software Requirements Definition Process and Products

2.1	Process.....	2-1
2.2	Products .....	2-1

### Section 3 – Reusability

3.1	Renaissance Building Blocks .....	3-1
3.2	CNMOS Projects .....	3-5

### Section 4 – Software Requirements

4.1	LPS System Requirements.....	4-1
4.1.1	System Overview .....	4-1
4.1.2	System Functional Overview .....	4-1
4.2	Programmatic Requirements.....	4-4
4.2.1	Development .....	4-4
4.2.2	Testing.....	4-5
4.2.3	Portability.....	4-5
4.3	Operational Requirements .....	4-5
4.3.1	User System Interface Requirements.....	4-5
4.3.2	Training .....	4-6
4.3.3	Maintenance.....	4-6
4.4	Raw Data Capture Subsystem.....	4-7
4.4.1	Functional Requirements .....	4-7
4.4.2	Performance Requirements.....	4-17
4.5	Raw Data Processing Subsystem.....	4-19
4.5.1	Functional Requirements .....	4-19
4.5.2	Performance Requirements.....	4-35
4.6	Major Frame Processing Subsystem .....	4-37
4.6.1	Functional Requirements .....	4-37
4.6.2	Performance Requirements.....	4-57
4.7	Payload Correction Data Subsystem.....	4-59
4.7.1	Functional Requirements .....	4-59
4.7.2	Performance Requirements.....	4-93

4.8	Image Data Processing Subsystem.....	4-95
4.8.1	Functional Requirements .....	4-95
4.8.2	Performance Requirements.....	4-111
4.9	Management and Control Subsystem.....	4-113
4.9.1	Functional Requirements .....	4-113
4.9.2	Performance Requirements.....	4-135
4.10	LPS Data Transfer Subsystem.....	4-137
4.10.1	Functional Requirements .....	4-137
4.10.2	Performance Requirements.....	4-149

## 5 – Database Analysis

5.1	Requirement Analysis and Conceptual Design .....	5-3
5.1.1	Functional Requirement Analysis .....	5-3
5.1.2	Performance Requirement Analysis.....	5-8
5.1.3	Programmatic Requirement Analysis.....	5-10
5.1.4	High-Level Entity Relationship Model .....	5-10

## Section 6 – User Interface

6.1	Task Analysis.....	6-1
6.1.1	Drivers.....	6-1
6.1.2	Constraints.....	6-2
6.1.3	Assumptions.....	6-2
6.1.4	Decisions.....	6-2
6.1.5	User Interface Event List .....	6-2
6.2	User Interface Goals .....	6-6
6.3	User Interface Mockup.....	6-6
6.3.1	Operating System.....	6-6
6.3.2	Reusability.....	6-6

## Appendix – Requirements Traceability

### Acronyms

### Figures

1-1	Landsat 7 Contact Periods Concept .....	1-4
1-2	LPS Files for Landsat 7 Contact Period 3.....	1-6
4-1	LPS Software Context-Level Diagram.....	4-2
4-2	LPS Level 0 Diagram.....	4-3
4-3	RDACS – DFD 1.0 .....	4-9
4-4	RDPS – DFD 2.0 .....	4-21
4-5	RDPS – DFD 2.1 .....	4-22

4-6	RDPS – DFD 2.2 .....	4-23
4-7	MFPS – DFD 3.0 .....	4-39
4-8	MFPS – DFD 3.3 .....	4-40
4-9	MFPS – DFD 3.4 .....	4-41
4-10	MFPS – DFD 3.5 .....	4-42
4-11	PCDS – DFD 4.0 .....	4-61
4-12	PCDS – DFD 4.1 .....	4-62
4-13	PCDS – DFD 4.6 .....	4-63
4-14	PCDS – DFD 4.7 .....	4-64
4-15	IDPS – DFD 5.0 .....	4-96
4-16	IDPS – DFD 5.2 .....	4-97
4-17	IDPS – DFD 5.4 .....	4-98
4-18	MACS – DFD 6.0 .....	4-115
4-19	MACS – DFD 6.9 .....	4-116
4-20	MACS – DFD 6.10 .....	4-117
4-21	LDTs – DFD 7.0 .....	4-139
5-1	LPS Database Design Approach .....	5-2
5-2	LPS ERD .....	5-13

---

## Section 1—Introduction

---

### 1.1 Scope

This software requirements specification (SRS) presents the detailed software requirements analysis for the Landsat 7 Processing System (LPS) software configuration components. The scope includes the functional, performance, operational, and programmatic requirements of the LPS.

The software requirements analysis is based on the LPS System Design Specification (SDS), LPS Functional and Performance Specification (F&PS), LPS Operations Concept document, and various technical studies completed by the LPS project.

This document is part of the LPS project baseline. It takes effect on approval by the Mission Operations and Systems Development Division Configuration Control Board (CCB). Proposed changes to this document require the same level of approval.

---

### 1.2 Applicable Documents

The documents listed in this section contain additional details regarding the LPS, the Landsat 7 System and project, and external systems.

---

#### 1.2.1 Specification Documents

The following documents provide the basis for developing the LPS software requirements presented in this specification:

1. Consultative Committee for Space Data Systems (CCSDS), CCSDS 701.0-B-1, *Recommendation for Space Data System Standards; Advanced Orbiting Systems (AOS), Networks and Data Links: Architectural Specification, Blue Book*, Issue 1, October 1989
2. National Aeronautics and Space Administration's (NASA's) Goddard Space Flight Center (GSFC), Mission Operations and Data Systems Directorate (MO&DSD), 560-8FPS/0194, *Landsat 7 Processing System (LPS) Functional and Performance Specification*, Revision 1, July 31, 1996

3. Martin Marietta Astro Space (MMAS), 23007702, *Landsat 7 System Data Format Control Book (DFCB), Revision D, Volume 4 – Wideband Data*, February 11, 1997
4. NASA/GSFC, *Interface Control Document (ICD) Between the Landsat Ground Station (LGS) and the Landsat 7 Processing System (LPS)*, Revision 1, October 17, 1996
5. —, 23007630, *Interface Control Document Between the Landsat 7 Processing System and the Image Analysis System (IAS)*, Revision 1, July 29, 1996
6. Computer Sciences Corporation (CSC), *SEAS System Development Methodology*, July 1989
7. NASA/GSFC, MO&DSD, 510-3FCD/0195, *Landsat 7 Processing System Output Files Data Format Control Book*, April 15, 1996
8. —, 560-3OCD/0194, *Landsat 7 Processing System (LPS) Operations Concept*, Revision 2, April 15, 1996
9. —, 514-3ICD/0195, *Landsat 7 Processing System Interface Definitions Document*, October 1996
10. —, 514-4DDS/0195, *Landsat 7 Processing System (LPS) As-Built Specification*, July 1997

---

### 1.2.2 Reference Documents

The following documents are used as sources of additional and background information, as required, for developing LPS software requirements:

1. NASA/GSFC, MO&DSD, MDOD-8YMP/0485, *Systems Management Policy*, July 1986
2. —, 430-L-000-2-A, *Landsat 7 System Specification*, Review, August 1994
3. —, *Landsat Ground Station (LGS) Functional and Performance Specification, Volume 1, Revision 1.0*, November 1994
4. —, 531-OCD-GS/Landsat 7, *Landsat Ground Station (LGS) Operations Concept*, November 1994
5. MMAS, Landsat 7 Library No. 5527, *Landsat 7 Image Assessment System (IAS) Operations Concept*, September 1994
6. NASA, *Landsat 7 Level 1 Requirements*, Draft, August 8, 1994

7. United States Geological Survey (USGS)/National Oceanic and Atmospheric Administration (NOAA), *Index to Landsat Worldwide Reference System (WRS) Landsats 1, 2, 3, and 4*, 1982
8. MMAS, 23007610A, *Landsat 7 System Program Coordinates System Standard*, proposed update draft, August 1994
9. NASA/GSFC, *LPS Software Management Plan*, August 1994
10. —, *Landsat 7 Detailed Mission Requirements*, January 1995
11. —, 430-11-06--003-0, *Landsat 7 System and Operations Concept (Pre-CCB Baseline Version)*, October 1994
12. Mission Operations and Data Systems Engineering, *Renaissance Catalog of Building Blocks*, Preliminary Draft, January 26, 1995

---

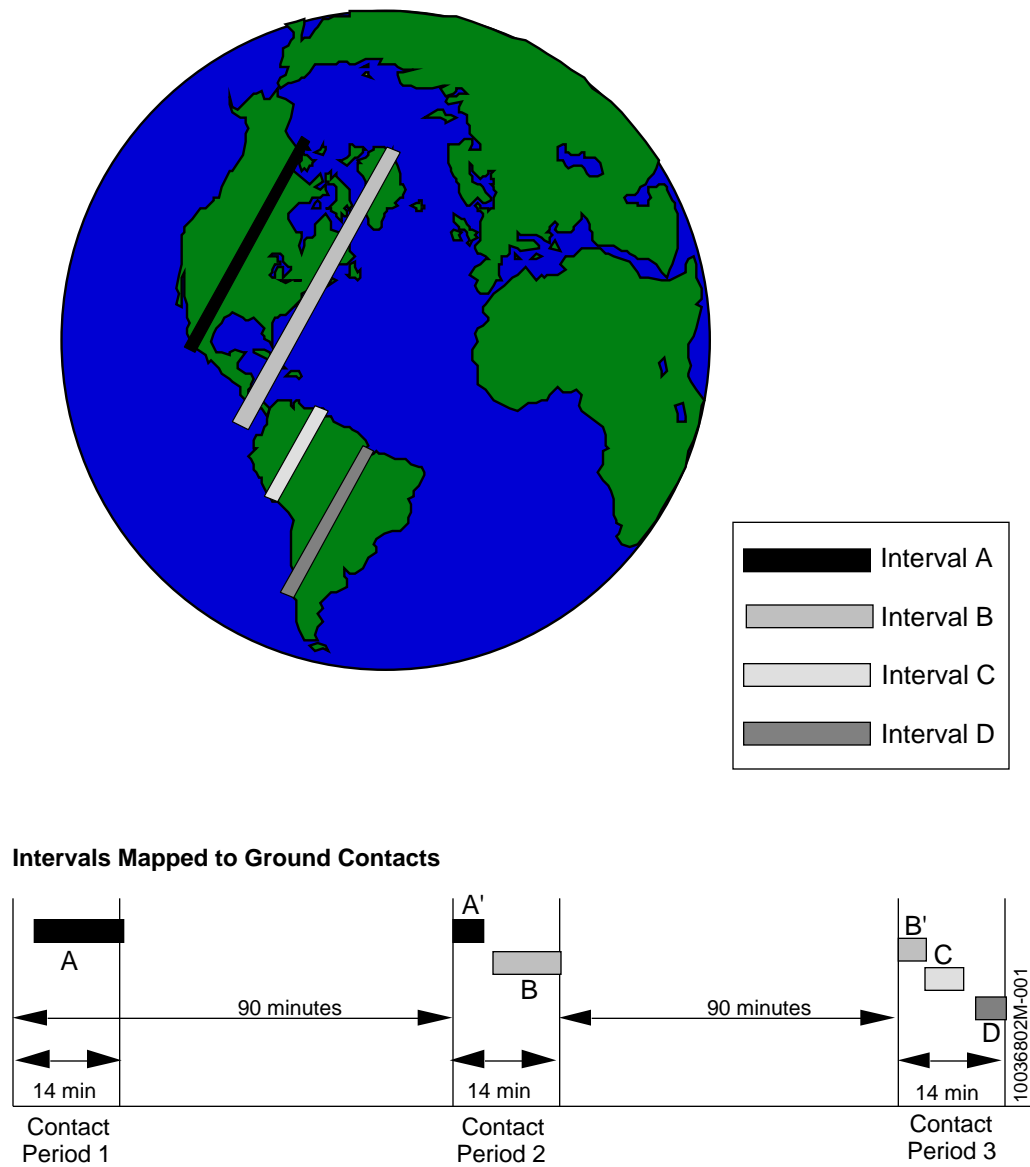
### 1.3 Definitions

The following terms are commonly used throughout this document to describe the LPS operations concept:

**Browse image file** – A reduced data volume file of the level-zero formatted (L0R) data that can be viewed to determine general ground area coverage and spatial relationships between ground area coverage and cloud coverage. This file contains browse image data from three predefined bands of the full-resolution Enhanced Thematic Mapper Plus (ETM+) format 1 scene data contained in the L0R instrument data files of a subinterval.

**Interval** – The time duration between the start and stop of an imaging operation (observation) of the Landsat 7 ETM+ instrument.

**Landsat 7 contact period** – The time duration between the start and end of a wideband data transmissions from the Landsat 7 spacecraft to a ground station. Figure 1–1 illustrates the Landsat 7 contact period concept.

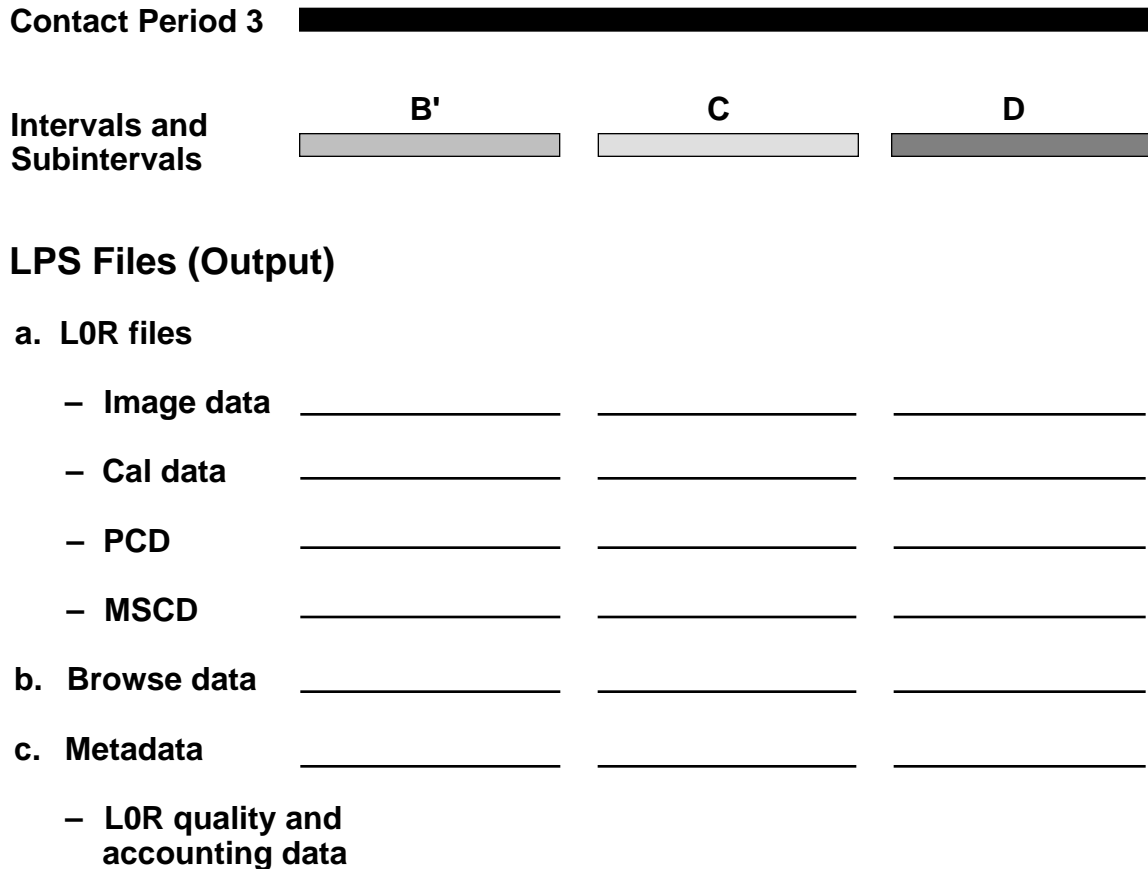


**Figure 1-1. Landsat 7 Contact Periods Concept**

**L0R files** – The reformatted, unrectified subinterval data having a sequence of pixels that are spatially consistent with the ground coverage and appended with radiometric calibration, attitude, and ephemeris data. Figure 1–2 illustrates the relationship of LPS files to the received subintervals.

- *Calibration file* – One file is created for each subinterval. This file contains all of the calibration data received on a major frame basis for a given subinterval. This is the data received after the scan line data [which follows the end of line (EOL) code] and before the next major frame sync, as described in *Landsat 7 System Data Format Control Book (DFCB), Revision A, Volume 4 – Wideband Data*. The data is grouped by detectors; for example, for a given major frame, detector 1 data is followed by detector 2 data. Reverse scans are reversed. The spacecraft time of the major frame corresponding to this data is appended, as well as the status data.
- *L0R instrument data file* – Each file contains the image data from a single band in a single subinterval. The data is grouped by detectors, for example, for a given major frame, detector 1 data is followed by detector 2 data. Reverse scan samples are changed to forward order. This data is nominally aligned using fixed and predetermined integer values that provide alignment for band offset, even and odd detectors, and forward and reverse scans. Quality indicators are stored in the database for each major frame and each subinterval.
- *Mirror scan correction data (MSCD)* – One file is created for each subinterval. This file contains the scan line data extracted from the two minor frames following the EOL code in each major frame of the subinterval. The scan line data includes the first half scan error (FHS ERR), second half scan error (SHS ERR), and scan direction (SCN DIR) information. The spacecraft time of the major frame corresponding to this data is appended.
- *Payload correction data (PCD)* – One file created for each subinterval. This file contains the PCD major frames received during a subinterval on a full PCD cycle basis. Quality indicators will be appended on a minor-frame basis.

**L0R quality and accounting data** – The data quality and accounting information collected by the LPS, on a subinterval basis, from processing of the ETM+ major frames constructed from the wideband virtual channel data units (VCDUs) received during a Landsat 7 contact period.



10036802M-002

**Figure 1–2. LPS Files for Landsat 7 Contact Period 3**

**Metadata** – One metadata file is created for each subinterval per string. The metadata contains information on the L0R data provided in the subinterval, and the names of the L0R instrument data, calibration data, PCD, MSCD, and browse image files associated with the subinterval. Metadata also contains quality and accounting information on the return-link wideband data used in generating the L0R file(s). In addition, metadata includes quality and accounting information on received and processed PCD and cloud cover assessment for the Worldwide Reference System (WRS) scene contained in the subinterval. The metadata is used to determine the subinterval and/or WRS scene level quality of the L0R data stored in the Land Processes Distributed Active Archive Center (LP DAAC).

**Return-link quality and accounting data** – The data quality and accounting information collected by the LPS from CCSDS Advanced Orbiting System (AOS) Grade 3 and Bose-Chaudhuri-Hocquenghem (BCH) error

detection and correction processing of the raw wideband data received from the Landsat Ground Station (LGS) on a Landsat 7 contact period basis.

**Subinterval** – A segment of raw wideband data interval received during a Landsat 7 contact period. Subintervals are caused by breaks in the wideband data stream due to communication dropouts and/or the inability of the spacecraft to transmit a complete observation (interval) within a single Landsat 7 contact period. The largest possible subinterval can be as long as a full imaging interval. The smallest possible subinterval can be as small as one full ETM+ scene with a time duration of approximately 24 seconds. Figure 1–1 illustrates the relationship between satellite on and off periods and satellite and ground contact periods.

---

## Section 2—Software Requirements Definition Process and Products

---

---

### 2.1 Process

The LPS software requirements analysis has been developed using *SEAS System Development Methodology* and tailored to suit the LPS project environment. LPS software requirements were analyzed by performing the following major activities:

1. Developing and analyzing an LPS software architecture that is based on LPS structured analysis, conforms to the selected hardware configuration and constraints, and maximizes the use of commercial off-the-shelf (COTS) items in its design.
2. Analyzing the LPS database, which is based on developing a conceptual and logical model of LPS data.
3. Analyzing a user interface for the LPS based on functional requirements and operations concepts.
4. Analyzing LPS system requirements using a computer-aided software engineering (CASE) tool—Cadre/Teamwork—that supports the structured analysis methodology.
5. Identifying LPS issues that, when resolved, may impact the LPS preliminary design.

---

### 2.2 Products

Several products are produced as a result of the software requirements analysis phase of the LPS. A model of the LPS exists in the Cadre/Teamwork CASE tool. This model includes data flow diagrams (DFDs), functional specifications, a data dictionary, and an entity relationship diagram for the LPS.

---

## Section 3—Reusability

---

### 3.1 Renaissance Building Blocks

A pool of software has already been developed and may be reused for the LPS. By identifying candidate software components, there is the potential to save time and money and to improve an existing component, instead of developing a component.

The following list of candidates for reuse include Renaissance Building Blocks and other MO&DSD projects. These candidates will be further investigated during the preliminary design phase of the LPS.

1. The Packet Extractor/Server (building block #RT-EX-05) removes the data unit zone (preferably a CCSDS packet structure versus a mission-unique structure ) of a CCSDS frame. It routes selected packets and quality annotations to its clients.
2. The Telemetry VCDU Statistics (building block #RT-EX-06) keeps reception statistics on all CCSDS AOS telemetry VCDUs received. It also extracts command link control word (CLCW) information from real-time VCDUs and maintains this information for use by commanding building blocks.
3. The Event Logger (building block #RT-HS-01) receives event messages from all real-time elements, time stamps them, and logs them into a history file. It provides an event server function that streams event messages to client processes based on specified filter parameters.
4. The History Logger (building block #RT-HS-02) logs annotated spacecraft telemetry frames and packets, command blocks, command echo blocks, Network Control Center (NCC) blocks, and Deep Space Network (DSN) monitor blocks to the history database manager. A header record is written to the log for each data block to ensure rapid access of data during replays and browsing.
5. The History Replay (building block #RT-HS-03) replays frame and packet history files back into the real-time data path so that real-time processing of the data can be repeated. While replaying frames, the output is sent to either the Packet Extractor/Server (CCSDS missions) or directly to Telemetry Decommuration (TDM

missions). Packets are always replayed directly to Telemetry Decommuration.

6. The Generic Equation Processor (building block #RT-TM-07) uses standard algebraic and trigonometric functions, derives values from the telemetry on the data server, and places the results back on the data server.
7. The Real-Time Attitude Determination (building block #RT-TM-08) processes real-time attitude sensor telemetry data to estimate the current spacecraft attitude using a Kalman filter and/or a single frame method. The system is flexible in terms of the sensor and dynamic models used, as well as which state parameters are estimated.
8. The Reports (building block #RT-US-05) generates American Standard Code for Information Interchange (ASCII) report files using data from real-time elements. It processes page snaps, sequential prints, event dumps, telemetry frame dumps, and telemetry packet dumps to an ASCII file, laser printer, or terminal emulator window.
9. The Event Printer (building block #RT-US-06) receives events from the event logger building block in real time and writes them to a dedicated line (events) printer.
10. The Data Distribution (building block #OL-DM-01) provides the capabilities for cataloging and distributing data products.
11. The Data Reception (building block #OL-DM-02) provides a method for receiving files from external interfaces. This software element polls a specified directory for new files at a user-specified interval. When a new file is received, the poller determines the file type and moves the file to a storage directory, updates the offline event log, and executes a script if further processing is required.
12. The File Services (building block #OL-DM-05) provides the capability for backup storage and retrieval and archival storage, retrieval, and access control of data files. The File Server Building Block backs up active, online data file, recovers files from backup media after data loss, transfers online data files to archival storage after a predetermined amount of time, and provides access to archived data files by authorized applications and users.
13. The Events Browser (building block #OL-DP-02) allows real-time and logged events information to be queried and displayed

according to several different filtering criteria. A graphical timeline view of event occurrences by type is provided, along with the traditional scrolling text window of event messages.

14. The Data Browser and Editor (building block #OL-DP-03) provides the capability to perform formatted hexadecimal octal, decimal, and ASCII dumps of all history data (e.g., transfer frame logs, packet logs, command blocks, command echo blocks, NCC blocks, DSN monitor blocks) and level-zero data sets. Dumps can be viewed on screen or routed to a disk or printer. Errors and missing data can be identified. Logged frames and packets can be edited for reprocessing by the level-zero processing (LZP) function. Another feature of the Data Browser and Editor is that it can display the availability of history data of different types within the time range selected.
15. The TDM Processor (building block #OL-DP-06) provides counts of minor, major, and incomplete major frames on a per-pass basis.
16. The Attitude Sensor Alignment and Calibration (building block #OL-SD-01) processes attitude history and attitude sensor telemetry data to estimate sensor alignment and calibration parameters using a batch filter algorithm. The system is flexible in terms of the sensor and dynamic models used, as well as which state parameters are estimated.
17. The Non-Real-Time Attitude Determination (building block #OL-SD-02) processes attitude sensor telemetry data offline to estimate the spacecraft attitude at a chosen epoch. In addition, the attitude may be propagated in either direction from the chosen epoch using a user-specified data parameter. The epoch attitude may be estimated by one of several methods, including batch least-squares or Kalman filter. The software is flexible in terms of the sensor and dynamic models used. Displays of uncertainties and residuals are provided so that the user may easily ascertain the validity of the solution.
18. The State Parameter Validation (building block #OL-SD-04) processes telemetry data to validate onboard computed parameters, including, but not limited to, gimbal angles, spacecraft attitude, orbit information, calibration parameters, and start observations. Statistics are computed including mean difference, root-mean-square residuals, and standard deviation.
19. The Attitude Measuring Processing (building block #OL-SD-05) processes converted telemetry data to obtain sensor and actuator

measurements adjusted for misalignment and bias in a user-specified reference frame. Resulting data include, but are not limited to, magnetic field vectors, spacecraft-to-celestial body vectors, and spacecraft body rates. The system is flexible in terms of data selection, reference frame, and output format.

20. The Session Manager (building block #OL-UI-01) provides the parameter editor capability to create, view, and modify data associated with an application. It also provides the Sequence Editor capability to sequentially execute a series of related applications.
21. The Display Builder (building block #OL-OU-02) allows the construction of new Motif graphical user interfaces (GUIs). The builder takes widgets from both the Motif library and user or project-specific libraries and puts these screen objects onto a palette. From the palette, these objects can be dragged to the interface under construction. All attributes of the screen objects can be modified before the file is saved as an industry-standard User Interface Language (UIL) file. Also, the preferred method for incorporating user-defined widgets into this tool is through an industry-standard Widget Meta Language definition.
22. The Network Time Source (building block #F-TS-01) is hardware and associated device level software and other related utilities used to provide synchronization to an external timing constant. Examples are various universal time coordinated (UTC) boards, NASA-36 boards, or WWV shortwave boards and antenna. It provides the interface to the required external timing source.
23. The Network Time Server (building block # F-TS-02) provides the authoritative source(s) of accurate network time for all other workstations and file servers. It typically includes the machine that incorporates the Network Timer Source interface. It also serves as a relay between the Network Timer Source and all other workstations and file servers so that they do not require an external interface.
24. The Network Time Client (building block #F-TS-03) queries the Network Timer Server for accurate network timing that is synchronized to the Network Timer Source.

---

## 3.2 CNMOS Projects

The Ground Operations Technology Testbed (Code 520) Distributed Process Control Program (DPCP) tool enables an operator to start a set of related processes running on one or more host computers. The sequence of steps and process dependencies is entered through use of a table that the operator sets up for the system. When so instructed, the DPCP starts all of the processes on their indicated machines in the proper order. In this way, the correct sequence of startups is consistently maintained and is executable each time the system is booted. The DPCP also continues to monitor each process and presents the status information to the operator on a graphical display, showing alert status when a process has died. The DPCP allows the operator to stop all the processes with a single command. This package is completely nonintrusive in that it does not need to be compiled or linked with any LPS software.

Data transfer software can be reused from the Packet Processor (Pacor) II and Data Distribution Facility (DDF) projects, both of which perform data transfer, deal with data availability notices (DANs), and use interprocess communication (IPC) to accomplish the data transfers.

The Ground Operations Technology Testbed (Code 520) Distributed Application Monitor Tool (DAMT) is helpful in analyzing a software system's performance. The DAMT may be used to monitor process on another machine, but its code must be compiled and/or linked with the application code. The DAMT currently cannot monitor processes with the same name on different machines in the same network.

The user interface portion (called the Integrator) of the Centralized Information System (CIS) for the Spacelab Data Processing Facility (SDPF) may be reused. The Integrator is a simple display that shows the status of pipeline processes and has a window that allows the operator to easily monitor system activities. This is a C/UNIX-based application that is portable. The message logging portion of the CIS also may be reused. This C/UNIX-based code uses embedded Structured Query Language (SQL) and is a structured and simple way of logging messages to the Integrator.

The Flight Dynamics Facility has potential software configuration items (SWCIs) that have potential use within the LPS. One SWCI computes the Greenwich hour angle from the Julian date. Another SWCI computes the geocentric inertial (GCI) Sun vector.

---

## Section 4—Software Requirements

This section describes LPS software requirements. These requirements are intended for LPS software executing on a single string. Overall system performance requirements have been divided down to the string level.

---

### 4.1 LPS System Requirements

---

#### 4.1.1 System Overview

The LPS software context-level diagram (Figure 4–1) shows the interactions the LPS software has with other LPS ground system components. Raw wideband data is accepted from the LGS, the LPS generates output files, and the LP DAAC is notified. The LPS operator issues directives to control the processing of the data. An external time source is used by five of the seven LPS subsystems to obtain the current system time.

LPS requirements from the F&PS have been allocated to each LPS subsystem. These requirements have been further divided between operations, hardware, and software. All software allocated has been analyzed to form this SRS.

Several requirements apply to all LPS subsystems. Refer to the “Requirements Assigned All Subsystems” section of Appendix A of the LPS SDS.

---

#### 4.1.2 System Functional Overview

The LPS Level 0 DFD (Figure 4–2) shows the internal flow of data between the seven LPS subsystems. The raw data flows into the raw data capture subsystem (RDCS), where it is captured. The raw data is then sent to the raw data processing subsystem (RDPS), where all transmission artifacts are removed, leaving spacecraft mission data. This mission data (Ann\_VCDU) is then passed to the major frame processing subsystem (MFPS), which interprets the mission data and passes PCD to the PCD processing subsystem (PCDS) and image data to the image data processing subsystem (IDPS). The management and control subsystem (MACS) accepts accounting information from the processing subsystems and creates a metadata file.





When the metadata file is created, the LPS data transfer subsystem (LDTS) notifies LP DAAC that the output files are ready for transfer. LP DAAC then is responsible for transferring the files from the LPS to LP DAAC. Once the files have been transferred, LDTS deletes them from LPS storage.

---

## **4.2 Programmatic Requirements**

---

### **4.2.1 Development**

The development environment for the LPS is the same for all LPS SWCIs. The goal of the LPS is to develop most functions using software [including application software, COTS, Government off-the-shelf (GOTS), system software]. Doing so will reduce the need for custom hardware and facilitate future upgrades in service capability.

The application software will be developed using C and UNIX on an Silicon Graphics, Inc. (SGI) Challenge series platform. All operating system upgrades must be coordinated with the LPS development team. All compilers and development tools must be upwardly compatible.

The ORACLE database management system (DBMS) COTS package will be used to manage the LPS database, generate reports (when applicable), and manage a user interface.

All Government-furnished equipment will remain available to the development team during the LPS design phases.

The LPS application software will be developed jointly by the LPS NASA developers and the LPS CNMOS developers at Goddard Space Flight Center (GSFC). The operational environment will be at the Earth Resources Observation System (EROS) Data Center (EDC) in Sioux Falls, South Dakota. Because the LPS development staff differs from the maintenance staff, the software must be developed and documented with this in mind. Online documentation, such as LPS-specific man pages, will be supported.

---

#### **4.2.2 Testing**

The LPS testing approaches are as follows:

- The system will be tested before delivery to EDC by LPS project personnel.
- The system will be acceptance tested at EDC by EDC personnel.
- Testing at GSFC will include simulated data from Generic Telemetry Simulator and spacecraft test data.
- The LPS will support ground system end-to-end testing.

---

#### **4.2.3 Portability**

The LPS application software needs to be upgradeable. For this reason, the application needs to be developed with portability in mind. POSIX-compliant code will be written whenever possible. Instances where code is not compliant due to performance requirements will be clearly documented to facilitate maintenance.

---

### **4.3 Operational Requirements**

LPS software will support and comply with the general operational requirements for the LPS.

---

#### **4.3.1 User-System Interface Requirements**

The user-system interface will be prototyped early during the LPS preliminary design phase. Sample screens and demonstrations will be provided to EDC personnel during that phase. Section 6 provides more details.

---

### **4.3.2 Training**

The training of the LPS operations personnel is **TBD**. Some training will be needed for both operations and maintenance personnel.

---

### **4.3.3 Maintenance**

LPS software will be maintained at the EDC by maintenance personnel. Maintenance personnel are not planned to be the same or any subset of development personnel.

---

## 4.4 Raw Data Capture Subsystem

The RDCS is responsible for receiving and capturing raw wideband data and placing it in a datastore for RDPS processing. The raw wideband data is saved on removable media with a tape label generated that corresponds to the saved raw wideband data. This subsystem also restages raw wideband data from the removable media to the online datastore for later processing, and for testing it can transmit raw data from one string to another. On request, RDCS generates a data receive summary.

---

### 4.4.1 Functional Requirements

The following functional requirements are allocated to the RDCS:

- Provide the capability to receive return-link wideband data for each contact period.
- Save return-link wideband data to removable media.
- Retrieve return-link wideband data on request.
- Generate a tape label for the specified raw wideband data.
- Generate an LPS wideband data receive summary for each contact period.
- Delete data for a specific contact period.
- Provide the capability to transmit/replay return-link wideband data through the LGS matrix switch.

#### 4.4.1.1 Major Functions

The RDCS captures a raw data byte stream after it receives the start capture directive from the MACS and outputs this raw wideband data to a datastore for further processing by other subsystems. The RDCS creates Contact\_Id information and notifies the MACS after a contact period ends. A contact summary report is generated following receipt of a directive from the MACS that includes the data set identifier.

The captured raw wideband data is saved following receipt of a MACS directive to start recording to the removable media. Also, a tape label for the saved raw wideband data file is generated. When a request to restage a contact period data set is issued by the MACS, the requested data set is recovered from the removable media to the online store.

When a MACS directive is received requesting a data receive summary, a report is generated describing the data set. The report includes data volume and approximate number of Landsat 7 scenes along with other identifiers.

The major functions of the RDCS are depicted in Figure 4–3.

#### 4.4.1.2 Detailed Functional Requirements

This section contains the process specifications for the lowest level processes in the RDCS DFDs.

NAME:

1.1

TITLE:

Receive Raw Wideband Data

INPUT/OUTPUT:

Raw\_WB\_Data : data\_inout

Contact\_Schedules : data\_inout

Raw\_WB\_Sets : data\_out

RDC\_Capture\_Stat : data\_out

RDC\_Acct : data\_out

RDC\_Capture\_Drct : data\_in

Current\_Time : data\_in

Raw\_Data\_Byte\_Stream : data\_in

BODY:

Description of Process

Receive a raw data serial stream and store as a raw wideband data set.

Assumptions

Preconditions

None

Postconditions

The Raw\_Data\_Byte\_Stream is placed into Raw\_WB\_Sets.

The raw wideband data accounting is placed into RDC\_Acct.

The RDC\_Capture\_Stat is sent to the MACS.

Constraints

RDC\_Capture\_Drct has two allowable values of "START" and "STOP". A value of "START" includes the following optional subdirectives:

Suspend Level OR Processing

Isolate and Restrict the Processor



Scheduled Capture Start Time  
Scheduled Capture Stop Time

#### Functional Breakdown

IF the RDC\_Capture\_Drct specifies to "START":

Isolate and restrict the processor if specified.

Suspend Level OR processing if specified.

Provide Raw\_WB\_Sets space availability via RDC\_Capture\_Stat.

Calculate Actual\_Start\_Time using Current\_Time.

Receive the Raw\_Data\_Byte\_Stream and store in a the Raw\_WB\_Data.

While receiving the Raw\_Data\_Byte\_Stream, provide progress status  
via the RDC\_Capture\_Stat.

Calculate Actual\_Stop\_Time using Current\_Time.

Define a Raw\_WB\_Set name for the Raw\_WB\_Data.

Calculate RDC\_Acct statistics.

Store the accounting, Raw\_WB\_Set name, and capture statistics into  
RDC\_Acct.

Delete the associated contact schedule from Contact\_Schedules.

Resume Level OR processing if it was suspended.

Deisolate and unrestrict the processor if isolated and restricted.

Provide capture success criteria via the RDC\_Capture\_Stat.

IF the RDC\_Capture\_Drct specifies to "STOP":

Terminate the current capture of the Raw\_Data\_Byte\_Stream.

Calculate Actual\_Stop\_Time using Current\_Time.

Define a Raw\_WB\_Set name for the Raw\_WB\_Data.

Calculate RDC\_Acct statistics.

Store the Raw\_WB\_Set name and capture statistics into RDC\_Acct.

Delete the associated contact schedule from Contact\_Schedules.

Resume Level OR processing if previously suspended.

Deisolate and unrestrict the processor if previously isolated and  
restricted.

Provide capture success criteria via the RDC\_Replay\_Stat.

#### Reusability

None.

NAME:

1.2

TITLE:

Save Raw Wideband Data

INPUT/OUTPUT:

Removable\_Media\_Configuration : data\_inout

RDC\_Generate\_Label\_Drct : data\_out

RDC\_Delete\_Drct : data\_out

RDC\_Acct : data\_out

Removable\_Media : data\_out

RDC\_Save\_Stat : data\_out

RDC\_Save\_Drct : data\_in

Raw\_WB\_Sets : data\_in

BODY:

Description of Process

Archive (save) a raw wideband data set to the removable media.

Assumptions

Preconditions

The Removable\_Media contains the media planned for archival.

Postconditions

The requested raw wideband data is placed into Removable\_Media.

The raw wideband data accounting is placed in RDC\_Acct.

RDC\_Save\_Stat is sent to the MACS.

Constraints

RDC\_Save\_Drct has two allowable values of "START" and "STOP." A value of "START" includes the Raw\_WB\_Set for archival, and the following optional subdirectives:

Removable\_Media slot number (if more than one available)

Functional Breakdown

IF the RDC\_Save\_Drct specifies to "START":

If the Removable\_Media slot number is specified, then

Use the specified value, otherwise, use the next available slot number from Removable\_Media\_Configuration.

Obtain the Raw\_WB\_Set from Raw\_WB\_Sets.

Archive the Raw\_WB\_Set to the Removable\_Media.

While archiving the Raw\_WB\_Set, provide progress status via the RDC\_Save\_Stat.

If the archiving was successful, then

Update RDC\_Acct to indicate successful archiving

Delete the Raw\_WB\_Set by sending a "START" directive via RDC\_Delete\_Drct

Generate a Media Label by sending a "START" directive via RDC\_Generate\_Label\_Drct

Place the next scheduled Removable\_Media slot number into Removable\_Media\_Configuration

Provide archival success criteria via the RDC\_Save\_Stat

IF the RDC\_Save\_Drct specifies to "STOP":

Terminate the current archiving of the Raw\_WB\_Set

Provide archival success criteria via the RDC\_Save\_Stat

Reusability

None.

NAME:

1.3

TITLE:

Restage Raw Wideband Data

INPUT/OUTPUT:

Raw\_WB\_Sets : data\_out

RDC\_Acct : data\_out

RDC\_Restage\_Stat : data\_out

RDC\_Restage\_Drct : data\_in

Removable\_Media : data\_in

BODY:

Description of Process

Restage a raw wideband data set from removable media.

Assumptions

Preconditions

The Removable\_Media contains the raw wideband data planned for restaging.

Postconditions

The raw wideband data on Removable\_Media is placed into Raw\_WB\_Sets.

The raw wideband data accounting is placed in RDC\_Acct.

The RDC\_Restage\_Stat is sent to the MACS.

Constraints

RDC\_Restage\_Drct has two allowable values of "START" and "STOP."

Functional Breakdown

IF the RDC\_Restage\_Drct specifies to "START":

Restage the Raw\_WB\_Set from the Removable\_Media

While restaging the Raw\_WB\_Set, provide progress status via the RDC\_Restage\_Stat

If the restaging was successful, then

Place the Raw\_WB\_Set into Raw\_WB\_Sets

Update RDC\_Acct to indicate successful archiving

Place the accounting statistics associated with the Raw\_WB\_Set into RDC\_Acct

Provide restaging success criteria via the RDC\_Restage\_Stat

IF the RDC\_Restage\_Drct specifies to "STOP":

Terminate the current restaging of the Raw\_WB\_Set

Remove the Raw\_WB\_Set

Provide restaging success criteria via the RDC\_Restage\_Stat

Reusability

None.

NAME:

1.4

TITLE:

Delete Raw Wideband Data

INPUT/OUTPUT:

RDC\_Acct : data\_inout

Raw\_WB\_Sets : data\_inout

RDC\_Delete\_Stat : data\_out

RDC\_Delete\_Drct : data\_in

BODY:

Description of Process

Delete a raw wideband data set.

Assumptions

Preconditions

The Raw\_WB\_Set to be deleted is in Raw\_WB\_Sets.

Postconditions

The Raw\_WB\_Set is deleted.

The RDC\_Delete\_Stat is sent to the MACS.

Constraints

RDC\_Delete\_Drct has one allowable value of "START". This includes the Raw\_WB\_Set to be deleted, and the following optional subdirectives:  
Unconditional Delete

Functional Breakdown

IF RDC\_Delete\_Drct specifies an Unconditional Delete:

Delete the specified Raw\_WB\_Set

Update RDC\_Acct to indicate the Raw\_WB\_Set was deleted

Provide deletion success criteria via the RDC\_Delete\_Stat

IF RDC\_Delete\_Drct does not specify an Unconditional Delete:

Obtain the processing status and archiving status from RDC\_Acct

If the Raw\_WB\_Set has been Level OR processed and has been archived, then

Delete the specified Raw\_WB\_Set

Update RDC\_Acct to indicate the Raw\_WB\_Set was deleted

Provide deletion success criteria via the RDC\_Delete\_Stat

Reusability

None.

NAME:

1.5

TITLE:

Replay Raw Wideband Data

INPUT/OUTPUT:

RDC\_Replay\_Stat : data\_out

Raw\_Data\_Byte\_Stream : data\_out

Raw\_WB\_Sets : data\_in

RDC\_Replay\_Drct : data\_in

BODY:

Description of Process

Replay a raw wideband data set as a raw data serial stream.

Assumptions

Preconditions

None.

Postconditions

The Raw\_WB\_Set is output as Raw\_Data\_Byte\_Stream.

The RDC\_Replay\_Stat is sent to the MACS.

Constraints

RDC\_Replay\_Drct has two allowable values of "START" and "STOP". A value of "START" includes the Raw\_WB\_Set for replay, and the following optional subdirectives:

Suspend Level OR Processing

Isolate and Restrict the Processor

Functional Breakdown

IF the RDC\_Replay\_Drct specifies to "START":

Isolate and restrict the processor if specified.

Suspend Level OR processing if specified.

Obtain the specified Raw\_WB\_Set from the Raw\_WB\_Sets

Replay the Raw\_WB\_Set as a serial stream through the  
Raw\_Data\_Byte\_Stream

While replaying the Raw\_WB\_Set, provide progress status via the  
RDC\_Replay\_Stat

Resume Level OR processing if it was suspended.

Deisolate and unrestrict the processor if isolated and restricted.

Provide replay success criteria via the RDC\_Replay\_Stat

IF the RDC\_Replay\_Drct specifies to "STOP":

Terminate the current replay of the Raw\_WB\_Set

Resume Level OR processing if previously suspended.

Deisolate and unrestrict the processor if previously isolated and  
restricted.

Provide replay success criteria via the RDC\_Replay\_Stat

Reusability

None.

NAME:

1.6

TITLE:

Generate Media Label

INPUT/OUTPUT:

RDC\_Media\_Label : data\_out

RDC\_Generate\_Label\_Stat : data\_out

RDC\_Generate\_Label\_Drct : data\_in

RDC\_Acct : data\_in

Removable\_Media\_Configuration : data\_in

BODY:

Description of Process

Generate a Media Label

Assumptions

Preconditions

RDC\_Acct contains accounting regarding the raw wideband data set associated with this label

Postconditions

The RDC\_Media\_Label is generated.

The RDC\_Generate\_Label\_Stat is sent to the MACS.

Constraints

RDC\_Generate\_Label\_Drct has one allowable value of "START."

This includes the Raw\_WB\_Set associated with the label, and the following optional subdirectives:

Removable\_Media slot number (if more than one available)

Functional Breakdown

If RDC\_Generate\_Label\_Drct specifies the Removable Media slot number, then

Use the specified value, otherwise use the next available slot number from Removable\_Media\_Configuration

Read the label parameters from RDC\_Acct

Output the RDC\_Media\_Label

Provide media label success criteria via the RDC\_Generate\_Label\_Stat

Reusability

None.

---

#### **4.4.2 Performance Requirements**

The following performance requirements are allocated to the RDCS:

- 4.4.2.1 The RDCS software on each LPS string shall generate the information necessary to produce a data receive summary for received wideband data within 10 seconds of the conclusion of its capture.
- 4.4.2.2 The RDCS software on each LPS string shall produce a data receive summary for the most recently received wideband data within 10 seconds of the receipt of an appropriate directive from MACS.
- 4.4.2.3 The RDCS software on each LPS string shall provide the capability to receive the equivalent of any combination of the format 1 and format 2 portions of 125 Landsat 7 ETM+ scenes of wideband data per day (approximately 25 to 30 gigabytes per day).
- 4.4.2.4 The RDCS software on each LPS string shall provide the capability to copy received wideband data to removable media at a minimum rate of 7.5 megabits per second (Mbps).
- 4.4.2.5 The RDCS software on each LPS string shall provide the capability to copy received wideband data to removable media concurrently with LOR processing of that data.
- 4.4.2.6 The RDCS software on each LPS string shall provide the capability to reprocess a maximum of 10 percent of a string's daily input volume of wideband data (approximately 12.5 scenes or 2.5 to 3 gigabytes per day).
- 4.4.2.7 The RDCS software on each LPS string shall provide the capability to copy received wideband data to removable media at a daily average aggregate rate of not less than 3 Mbps (includes 10 percent of overhead due to reprocessing).
- 4.4.2.8 The RDCS software on each LPS string shall maintain data processing throughput performance for all Landsat 7 raw wideband data received with a bit error rate (BER) of one bit error

in  $10^5$  bits, without loss of LOR processed data and without retransmission.

- 4.4.2.9 The RDCS software on each LPS string shall provide the capability of receiving wideband data from a single LGS output channel at a maximum rate of 75 Mbps.
- 4.4.2.10 The RDCS software on each LPS string shall provide the capability to receive wideband data for Landsat 7 contact periods of up to 14 minutes.
- 4.4.2.11 The RDCS software on each LPS string shall output any periodic processing status information that it generates with a maximum latency of 30 seconds between output.

---

## 4.5 Raw Data Processing Subsystem

The RDPS is responsible for processing the raw wideband data on a contact period basis, including synchronizing the frames, performing various error detection and correction techniques on the data, and, on request, generating a report on the quality of the data.

---

### 4.5.1 Functional Requirements

The following functional requirements are allocated to the RDPS:

- Validate and store the processing parameters received from MACS.
- Process the wideband data on a contact period basis.
- Detect and synchronize on normal and inverted polarity wideband data concurrently.
- Use a search/check/lock/flywheel (SCLF) strategy for synchronization.
- Invert bits with inverted polarity.
- Correct bit slips.
- Perform pseudorandom noise (PRN) decoding.
- Perform CCSDS AOS Grade 3 service.
- Store all channel access data units (CADUs) that have failed the CCSDS AOS grade 3 checks.
- Annotate VCDUs containing fill data.
- Perform BCH error detection and correction.
- Store all CADUs that have failed BCH error detection and correction on the mission data zone.
- Generate a return-link quality and accounting report on a Landsat 7 contact period basis.
- Annotate the CADU with the virtual channel identifier (VCID) change information.

#### 4.5.1.1 Major Functions

The RDPS first checks the CCSDS parameters for valid values and stores the parameters if they are valid. If the CCSDS parameters are invalid, default values will be used.

The raw wideband data is retrieved for the contact period requested by the operator. The frames are then synchronized using the SCLF strategy. Following SCLF synchronization, the frames are aligned on byte boundaries. If the frame synchronization pattern is inverted, the data is normalized reversing the data inversion. The data is then decoded to reverse the PRN encoding.

Grade 3 error detection is performed including a cyclic redundancy check (CRC) calculation and Reed-Solomon (RS) checks on the header. Any fill CADUs are annotated.

The final error detection performed on the data is the BCH decode process that is performed on the mission and pointer data fields. This process also attempts to correct the data. Copies of the CADUs that fail the error detection processes are saved. The VCDUs are annotated to reflect the data quality and any change in the VCID. They are also annotated to mark the end of the contact period. Information pertaining to the quality of the data is stored to be used in generating the return-link quality and accounting report.

The major functions of the RDPS are depicted in Figures 4–4 through 4–6.

#### 4.5.1.2 Detailed Functional Requirements

This section contains the process specifications for the lowest level processes in the RDPS DFDs.

NAME:

2.1.1

TITLE:

Perform SCLF Sync

INPUT/OUTPUT:

Chan\_Acss\_Acct : data\_inout

Contact\_Id : data\_out

RDP\_Sync\_Err\_Status : data\_out

Sync\_WB\_Data : data\_out

Valid\_CCSDS\_Parms : data\_in

Raw\_Data\_Byte\_Stream : data\_in

RDP\_Process\_Drct : data\_in







Sync\_Thres : data\_in

#### BODY:

##### Description of Process

This process uses the search/check/lock/flywheel strategy to locate frame sync patterns in the Raw\_Data\_Byte\_Stream.

##### Assumptions

###### Preconditions

Valid\_CCSDS\_Parms and the RDP\_Process\_Drct have been received.

###### Postconditions

Sync\_WB\_Data which contains the Contact\_Id, Raw\_Data\_Byte\_Stream, frame sync marker locations, and Sync\_Annotation have been sent to Align Bytes.

MACS has been notified if the cumulative number of frame sync errors have exceeded the tolerance stored in Valid\_RDP\_Thres.Sync\_Thres

The RDP\_Acct contains NULL fields for the specified Contact\_Id for all return-link QA information not provided by this process

###### Constraints

The expected frame sync pattern (1acffc1d) must exist in the data stream and must be spaced a CADU length apart, within bit slip or flywheel tolerance range.

##### Functional Breakdown

Retrieve the Raw\_Data\_Byte\_Stream associated with the Contact\_Id specified in the RDP\_Process\_Drct from the Raw\_WB\_Sets

Place the Contact\_Id into the Sync\_WB\_Data.Contact\_Id.

Retrieve the Valid\_CCSDS\_Parms.

Utilize the search/check/lock/flywheel strategy to locate the valid normal and inverted frame synchronization markers Sync\_WB\_Data.

Polarity\_Unknown\_Sync.Sync and Sync\_WB\_Data.Polarity\_Unknown\_Sync.Inverted\_Sync.

Detect Sync\_WB\_Data.Polarity\_Unknown\_Sync.Sync and Sync\_WB\_Data.

Polarity\_Unknown\_Sync.Inverted\_Sync with  $\pm 3$  bit slips, and fill or truncate the frame length accordingly

After each valid synchronization pattern is located, provide the synchronization marker exact location in Sync\_WB\_Data.Sync\_Loc along with the Sync\_WB\_Data.Sync\_Annotation.

Set the Ann\_VCDU.End\_Of\_Contact\_Flag (set to TRUE for the final BCH\_Chkd\_VCDU for the contact period) to indicate the end of the contact period.

During all modes allow the error tolerance indicated in

Valid\_CCSDS\_Parms.CADU\_Sync\_Lock\_Error\_Tol,  
Valid\_CCSDS\_Parms.CADU\_Sync\_Marker\_Check\_Error\_Tol,  
Valid\_CCSDS\_Parms.CADU\_Bit\_Slip\_Correction\_Extent

During all modes accumulate

Chan\_Acss\_Acct.Inverted\_CADU\_Cnt,  
Chan\_Acss\_Acct.Polarity\_Change\_Cnt,  
Chan\_Acss\_Acct.CADU\_Bit\_Slip\_Cnt,  
Chan\_Acss\_Acct.CADU\_Sync\_Err\_Cnt,  
Chan\_Acss\_Acct.CADU\_Rcv\_Cnt, and

Chan\_Acss\_Acct.CADU\_Flywheel\_Cnt  
Insert the Valid\_CCSDS\_Parms into the Chan\_Acss\_Acct.Valid\_CCSDS\_Parms.  
Place the RDP\_Process\_Drct.Contact\_Id into Chan\_Acss\_Acct.Contact\_Id.  
Place the Chan\_Acss\_Acct into the RDP\_Acct store identified by the  
Chan\_Acss\_Acct.Contact\_Id.  
For every cumulative error detected, compare with the threshold value stored in  
Valid\_RDP\_Thres.Sync\_Thres.  
If the number of cumulative errors per contact period exceeds the threshold, then  
Send a RDP\_Status.RDP\_Sync\_Err\_Status to the MACS indicating the  
tolerance has been exceeded.  
Output the RDP\_Process\_Drct.Contact\_Id  
Output the Sync\_WB\_Data.

#### Reusability

A frame sync lookup table would allow bit flips in addition to bit slips.  
A prototype exists for portions of this process which will be examined for  
reuse.

NAME:

2.1.2

TITLE:

Align Bytes

INPUT/OUTPUT:

Aln\_CADU : data\_out

Aln\_Inver\_CADU : data\_out

Sync\_WB\_Data : data\_in

BODY:

Description of Process

This process takes each frame of data starting with the first frame sync pattern detected and aligns the entire frame on a byte boundary.

Assumptions

Preconditions

The location of the frame sync patterns is known. The frame size is known.

The raw wideband data has been extracted on a contact period basis.

Postconditions

Every CADU frame has been aligned on a byte boundary. Fill data is used if the frame size is not an even multiple of the increment being shifted.

Constraints

None

Functional Breakdown

Obtain the location of the frame sync pattern from the Sync\_WB\_Data.

Calculate the distance needed to shift the data.

Shift the entire frame to align on a byte boundary

Complete the Aln\_CADU by adding fill data to the remainder of the frame.

If the frame sync pattern is inverted, then

Place Sync\_WB\_Data.Contact\_Id into Aln\_Inver\_CADU.Contact\_Id.

Output the Aln\_Inver\_CADU,

Otherwise

Place Sync\_WB\_Data.Contact\_Id into Aln\_CADU.Contact\_Id.

Output Aln\_CADU.

Reusability

A prototype exists for this process which will be examined for reuse.

NAME:

2.1.3

TITLE:

Deinvert Data

INPUT/OUTPUT:

Aln\_CADU : data\_out

Aln\_Inver\_CADU : data\_in

BODY:

Description of Process

This process flips the bits of an entire CADU.

Assumptions

Preconditions

The frame sync pattern for the CADU is inverted.

Postconditions

The entire CADU is deinveterd (normalized).

Constraints

None.

Functional Breakdown

Flip all bits of the Aln\_Inver\_CADU(0->1, 1->0) to create the bits of the Aln\_CADU.

Place the Aln\_Inver\_CADU.Contact\_Id into Aln\_CADU.Contact\_Id.

Reusability

A prototype exists for this process and it will be examined for reuse.

NAME:

2.1.4

TITLE:

Perform PN Decode

INPUT/OUTPUT:

Ann\_CADU : data\_out

Aln\_CADU : data\_in

BODY:

Description of Process

This process decodes the data to reverse the PN encoding on the data.

Assumptions

Preconditions

CADUs have been byte aligned in the Align Bytes process.

Inverted data has been normalized in the Deinvert Data process.

Postconditions

Data is PN decoded

Constraints

None.

Functional Breakdown

The Aln\_CADU is PN decoded using a standard pseudo-random sequence generated by a polynomial described in CCSDS 101.0-B-3, paragraphs 6.3, 6.4, and 6.5 to transform the Aln\_CADU to an Ann\_CADU.

Place the Aln\_CADU.Contact\_Id into Ann\_CADU.Contact\_Id.

Output the Ann\_CADU.

Reusability

A prototype exists for this function which will be considered for reuse.

NAME:

2.2.1

TITLE:

Perform CRC Check

INPUT/OUTPUT:

CRC\_Acct : data\_inout

RDP\_CRC\_Err\_Status : data\_out

CRC\_Failed\_CADU : data\_out

CRC\_Chkd\_CADU : data\_out

Ann\_CADU : data\_in

CRC\_Thres : data\_in

BODY:

Description of Process

This process performs a 16-bit Cyclic Redundancy Check on the CADU

Assumptions

Preconditions

The data has been synchronized, byte aligned, CRC encoded, and PN decoded.

Postconditions

CRC errors are reported to the CRC\_Acct.

Constraints

None

Functional Breakdown

Perform a Cyclic Redundancy Check on the Ann\_CADU data using the generator polynomial and procedure described in CCSDS 701.0-B-1, Section 5.4.9.2.1.4.2.c

Place the Ann\_CADU.Contact\_Id into the CRC\_Acct.Contact\_Id.

Update the RDP\_Acct store identified by the CRC\_Acct.Contact\_Id with the accumulated CRC\_Acct information.

If the CADU fails the CRC check, then

Place the CRC\_Failed\_CADU into the Failed\_CADUs store identified by Ann\_CADU.Contact\_Id.

For every CRC error detected,

Fetch the threshold value stored in Valid\_RDP\_Thres.CRC\_Thres.

If the cumulative number of CRC errors per contact period threshold value is exceeded, then

Send a RDP\_Status.RDP\_CRC\_Err\_Status message to the MACS indicating that the Valid\_RDP\_Thres.CRC\_Thres tolerance has been exceeded. Output the CRC\_Chkd\_CADUs.

Reusability

A prototype of this process is being developed and will be examined for reuse.

NAME:

2.2.2

TITLE:

Perform RS\_EDAC Check

INPUT/OUTPUT:

RS\_Acct : data\_inout

RDP\_RS\_Err\_Status : data\_out

RS\_Failed\_CADU : data\_out

RS\_Corr\_CADU : data\_out

CRC\_Chkd\_CADU : data\_in

RS\_Thres : data\_in

BODY:

Description of Process

This process performs Reed Solomon error detection and correction on the VCDU header.

Assumptions

Preconditions

Data has been synchronized, byte aligned, PN decoded, and RS encoded

Postconditions

If errors are detected and the error tolerance for Reed Solomon has not been exceeded, the VCDU header is considered corrected.

Constraints

None

Functional Breakdown

Receive CRC\_Chkd\_CADU and use bits 48 through 63 (16 bits) of the VCDU header as the check symbols of the shortened Reed-Solomon (10,6) code.

If the VCDU\_Hdr\_Bytes contained within CRC\_Chkd\_CADU.VCDU\_Bytes are uncorrectable, then

Place the RS\_Failed\_CADU into the Failed\_CADUs store identified by the CRC\_Chkd\_CADU.Contact\_Id.

Accumulate the RS\_Acct.VCDU\_Header\_Uncorrectable\_Err\_Cnt.

Otherwise,

Place CRC\_Chkd\_CADU.Contact\_Id into the RS\_Corr\_CADU.Contact\_Id.

If this is a fill VCDU, then

Place the VCDU\_Fill\_Hdr\_Bytes in the RS\_Corr\_CADU

Otherwise

Place the VCDU\_Hdr\_Bytes in the RS\_Corr\_CADU.

The remaining data portion of the VCDU will be identified as the RS\_Corr\_CADU.VCDU\_Data.

Place the RS\_Annotation into the RS\_Corr\_CADU.RS\_Annotation.

If the RS\_Corr\_CADU has a VCID corresponding to format 1, then

Accumulate the RS\_Acct.VCDU\_Header1\_Correctable\_Err\_Cnt

If the RS\_Corr\_CADU has a VCID corresponding to format 2, then

Accumulate the RS\_Acct.VCDU\_Header2\_Correctable\_Err\_Cnt

Place the RS\_Corr\_CADU.Contact\_Id into RS\_Acct.Contact\_Id.

Update the RDP\_Acct store identified by the RS\_Acct.Contact\_Id with the accumulated RS\_Acct information.

For every uncorrectable header detected, fetch the threshold value stored in  
Valid\_RDP\_Thres.RS\_Thres.

If the cumulative number of RS errors per contact period threshold value is  
exceeded, then

Send a RDP\_Status.RDP\_RS\_Err\_Status message to the MACS stating that the  
Valid\_RDP\_Thres.RS\_Thres has been exceeded.

Output the RS\_Corr\_CADU.

Reusability

None.

NAME:

2.3

TITLE:

Decode BCH

INPUT/OUTPUT:

BCH\_Acct : data\_inout

RDP\_BCH\_Err\_Status : data\_out

BCH\_Chkd\_VCDU : data\_out

BCH\_Failed\_VCDU : data\_out

Grade\_3\_Chkd\_VCDU : data\_in

BCH\_Thres : data\_in

BODY:

Description of Process

This process checks the codewords of the VCDU mission and pointer data for bit errors and attempts to correct the errors.

Assumptions

Preconditions

This process will only be executed if there are any bit slips, frame sync errors, Reed-Solomon errors, or CRC errors in the Grade\_3\_Chkd\_VCDU.

Postconditions

The VCDU has been further annotated with the mission and pointer field check quality indicators.

A copy of the VCDUs which failed the BCH EDAC on the mission data zone has been saved.

Constraints

None

Functional Breakdown

Decode the mission data zone and data pointer of the Grade\_3\_Chkd\_VCDU which have been encoded by the polynomials defined in the Landsat 7 System Data Format Control Book, sections 3.1.2.1.2 and 3.1.2.1.3.

Attempt to correct the bit errors.

Place the Grade\_3\_Chkd\_VCDU.Contact\_Id into the BCH\_Chkd\_VCDU.Contact\_Id.

Place the Grade\_3\_Chkd\_VCDU data into the BCH\_Chkd\_VCDU

If the bits are BCH corrected, then

Insert the corrected bits into the BCH\_Chkd\_VCDU.

BCH\_Corrected\_Data

Accumulate the

BCH\_Acct.BCH\_Data\_Corrected\_Err\_Cnt,

BCH\_Acct.BCH\_Pointer\_Corrected\_Err\_Cnt,

BCH\_Acct.BCH\_Data\_Uncorrected\_Err\_Cnt,

BCH\_Acct.BCH\_Pointer\_Uncorrected\_Err\_Cnt,

BCH\_Annotation.BCH\_Mission\_Bits\_Corrected\_Count, and

BCH\_Annotation.BCH\_Pointer\_Bits\_Corrected\_Count

If the error counts exceed the Valid\_RDP\_Thres.BCH\_Thres, then

Send the RDP\_Status.RDP\_BCH\_Err\_Status containing the number of BCH errors to the MACS.

If the VCDU is uncorrectable for the mission data zone, then  
    Output the BCH\_Failed\_VCDU to the Failed\_CADUs store identified by the  
    Grade\_3\_Chkd\_VCDU.Contact\_Id.  
Place the BCH\_Chkd\_VCDU.Contact\_Id into BCH\_Acct.Contact\_Id.  
Update the RDP\_Acct store identified by the BCH\_Acct.Contact\_Id with the  
    accumulated BCH\_Acct information.  
Place the BCH\_Data\_Quality\_Indicator into the BCH\_Annotation.  
    BCH\_Mission\_Quality\_Indicator to indicate that the mission data zone is  
    “CORRECTABLE”, “UNCORRECTABLE”, or has “NO\_ERRORS”.  
Place the BCH\_Pointer\_Quality\_Indicator into the BCH\_Annotation.  
    BCH\_Pointer\_Quality\_Indicator to indicate that the pointer data zone is  
    “CORRECTABLE”, “UNCORRECTABLE”, or has “NO\_ERRORS”.  
Append the BCH\_Annotation to the BCH\_Chkd\_VCDU.  
Output the BCH\_Chkd\_VCDU.

#### Reusability

See the Landsat 7 Mission Data and Data Pointer BCH Decoder Prototype  
Description for possible reuse of the prototype.

NAME:

2.4

TITLE:

Annotate VCID Change

INPUT/OUTPUT:

Curr\_VCID : data\_out

Ann\_VCDU : data\_out

BCH\_Chkd\_VCDU : data\_in

Prev\_VCID : data\_in

Contact\_Id : data\_in

BODY:

Description of Process

This process determines if there is a change in the virtual channel of the VCDU and annotates the Ann\_VCDU to indicate the change.

Assumptions

Preconditions

Frame sync, CRC and Reed-Solomon checks have been performed on the VCDUs.

The previous VCID has been retained.

Postconditions

The VCDU has been annotated with a VCID change flag.

Constraints

None.

Functional Breakdown

Place the BCH\_Chkd\_VCDU.Contact\_Id into the Ann\_VCDU.Contact\_Id.

Place the BCH\_Chkd\_VCDU information into the Ann\_VCDU.

If Curr\_VCID is Fill\_VCID

Set the Ann\_VCDU.Fill\_CADU\_Flag to TRUE

Increment the Fill\_CADU\_Cnt

If the Saved\_VCID.Prev\_VCID differs from the Curr\_VCID, then

Set the Ann\_VCDU.VCID\_Change\_Flag to TRUE to indicate a change in the VCID.

Otherwise,

Set the Ann\_VCDU.VCID\_Change\_Flag to FALSE to indicate no change in the VCID.

Output the Curr\_VCID to the Saved\_VCID datastore.

Output the Ann\_VCDU.

Reusability

None.

---

## 4.5.2 Performance Requirements

The RDPS has the following performance requirements:

- 4.5.2.1 The RDPS software on each LPS string shall provide the capability to process the equivalent of any combination of the format 1 and format 2 portions of 125 Landsat 7 ETM+ scenes of wideband data per day (approximately 25 to 30 gigabytes per day).
- 4.5.2.2 The RDPS software on each LPS string shall process received wideband data at a minimum rate of not less than 7.5 Mbps (based on a peak raw wideband throughput of 7.5 Mbps).
- 4.5.2.3 The RDPS software on each LPS string shall provide the capability to execute concurrently with all other LPS subsystems.
  - 4.5.2.3.1 The RDPS software shall begin to process received raw wideband data immediately on receipt of required input.
  - 4.5.2.3.2 The RDPS software shall output the equivalent of one Landsat 7 ETM+ scene (215,445 CADUs) within 250 seconds of the time either at the beginning of processing or the time of its last output.
- 4.5.2.4 The RDPS software on each LPS string shall provide the capability to reprocess a maximum of 10 percent of a string's daily input volume of wideband data (approximately 12.5 scenes or 2.5 to 3 gigabytes per day).
- 4.5.2.5 The RDPS software on each LPS string shall provide the capability to process received wideband data at a daily average aggregate rate of 3 Mbps (includes 10 percent of overhead due to reprocessing).
- 4.5.2.6 The RDPS software on each LPS string shall maintain data processing throughput performance for all Landsat 7 raw wideband data received with a BER of one bit error in  $10^5$  bits, without loss of level-zero processed data and without retransmission.

- 4.5.2.7 The RDPS software on each LPS string shall provide the capability to retrieve retained wideband data at rates equal to or greater than 7.5 Mbps.
- 4.5.2.8 The RDPS software on each LPS string shall output any periodic processing status information that it generates with a maximum latency of 30 seconds between output.

---

## **4.6 Major Frame Processing Subsystem**

The MFPS is responsible for identifying a major frame and determining the major frame time. This subsystem determines the subinterval from the major frame time, the VCID, or the contact period; generates calibration and MSCD LOR files on a subinterval basis and scene data and PCD data on a major frame basis; and generates all LOR quality and accounting information.

---

### **4.6.1 Functional Requirements**

The following functional requirements are allocated to the MFPS:

- Identify and collect VCDUs on a major frame basis.
- Locate the minor frames that contain the major frame synchronization, major frame time, and EOL codes, scene data, calibration data, and MSCD.
- Extract PCD on a major frame basis data
- Identify the number of missing VCDUs.
- Determine the major frame time.
- Determine the start and end of a subinterval.
- Deinterleave and align wideband data on a major frame basis.
- Extract, process, and generate calibration and MSCD files on a subinterval basis.
- Collect and generate LOR quality and accounting information on subinterval and major frame basis.

#### **4.6.1.1 Major Functions**

The following paragraphs summarize the major functions.

In the Identify VCDUs process, VCDUs are collected for one major frame such that all VCDUs have the same contact identifier and VCID, increasing minor frame counters, and no data break. VCDU and minor frame counters

are checked. All VCDUs that do not qualify to be part of a major frame set are put into a trouble file.

The Extract PCD process receives the VCDU in both a major frame set and in the trouble file and extracts the PCD bytes in the VCDUs. The extracted PCD bytes are sent to the PCDS.

The Parse Major Frame process is responsible for locating, extracting, and validating the major frame synchronization and the EOL codes and the six major frame time minor frames. Start and end of subintervals is detected. LOR quality and accounting information is aggregated on a major frame basis.

During the Generate Band Data process, wideband data is deinterleaved and reversed. Band alignment takes place according to the sensor alignment information. In the case of missing major frames, the MFPS outputs aligned bands containing fill band data.

The Extract Calibration and MSCD process extracts calibration and MSCD data and generates the calibration and MSCD LOR files.

The Collect Quality and Accounting process outputs the LOR quality and accounting information aggregated on a major frame basis received from the Parse Major Frame process. LOR quality and accounting information aggregated on a major frame basis is also aggregated on a subinterval basis and stored. The major functions of MFPS are depicted in Figures 4–7 through 4–10.

#### **4.6.1.2 Detailed Functional Requirements**

This section contains the process specifications for the lowest level processes in the MFPS DFDs.

NAME:

3.1

TITLE:

Identify VCDUs

INPUT/OUTPUT:

Ann\_VCDU\_Collection : data\_inout

VCDU\_With\_Fill\_Info : data\_out

Major\_Frame\_VCDU\_Set : data\_out

Sub\_Intv\_Info : data\_out

Scan\_Dir\_Thres : data\_in

Ann\_VCDU : data\_in









**BODY:****Description of Process**

Collect Ann\_VCDUs on a major frame basis and check the VCDU counters.

**Assumptions****Preconditions**

Receive one Ann\_VCDU at a time.

The Ann\_VCDU.Sync\_Annotation.End\_Of\_Contact\_Flag indicates the last VCDU of the current contact period.

The VCDU quality fields of the Ann\_VCDU are set.

**Post conditions**

Major\_Frame\_VCDU\_Set, a set of VCDUs with the same VCID and same contact id, and increasing mnf counters is sent to Parse Major Frame. The first and last VCDU of the set may not have the same attributes as the rest.

Missing VCDUs are detected and flagged in Major\_Frame\_VCDU\_Set.

All VCDUs which do not qualify to be part of a Major\_Frame\_VCDU\_Set are put in the Failed\_Mjf\_Data store.

**Constraints**

None.

**Functional Breakdown**

Receive Ann\_VCDUs one at a time until Major\_Frame\_VCDU\_Set is found.

Discard it if it is a fill CADU.

Check the correctness of VCDU and mnf counters of each VCDU.

Collect a set of VCDUs with the same VCID, same contact id and increasing mnf counters without a data break into Major\_Frame\_VCDU\_Set.

The set starts and ends with a pair of VCDUs having the mnf counter drop as the VCDU counter increases.

Collect all VCDUs which do not qualify to be part of a Major\_Frame\_VCDU\_Set into the Failed\_Mjf\_Data store.

Stop collection of VCDUs if the End\_Of\_Contact\_Flag indicates the last VCDU of the current contact period or the VCID changes.

Fill the missing VCDUs with fill data and annotate both Major\_Frame\_VCDU\_Set and Failed\_Mjf\_Data with missing VCDU information.

**Reusability**

Prototypes should be of use for check on the VCDU sequence count.

NAME:

3.2

TITLE:

Extract PCD

INPUT/OUTPUT:

Status\_Info : data\_out

PCD\_Info : data\_out

VCDU\_With\_Fill\_Info : data\_in

Sub\_Intv\_Id : data\_in

BODY:

Description of Process

Extract PCD bytes from status field of the VCDU

Assumptions

Preconditions

The number of missing VCDUs must be provided with  
Major\_Frame\_VCDU\_Set

Postconditions

PCD\_Info contains the four extracted PCD bytes, the number of missing  
VCDUs, and the end of contact marker.

Constraints

None.

Functional Breakdown

Extract words #1 through #4 of the PCD/Status data from each non-fill VCDU in the  
Major\_Frame\_VCDU\_Set and place into PCD\_Info.

Extract words #1 through #4 of the PCD/Status data from each non-fill VCDU in the  
Failed\_Mjf\_Data and place into PCD\_Info.

Annotate the PCD\_Info with missing VCDU information, End\_Of\_Contact\_Flag and  
Sub\_Intv\_Id

Extract words #7 and #8 of the PCD/Status data from each non-fill VCDU in the  
Major\_Frame\_VCDU\_Set and place into Status\_Info.

Output Status\_Info and PCD\_Info.

Reusability

None.

NAME:

3.3.1

TITLE:

Identify Major Frames

INPUT/OUTPUT:

Mjf\_QA : data\_inout

Mjf\_VCDU\_Data : data\_out

Failed\_Mjf\_Data : data\_out

Time\_Code : data\_out

Major\_Frame\_VCDU\_Set : data\_in

Maj\_Vote\_Tol : data\_in

Sub\_Intv\_Info : data\_in

Part\_Mnf\_Tol : data\_in

BODY:

Description of Process

Major\_Frame\_VCDU\_Set is searched for the major frame synchronization and the end of line codes.

Assumptions

Preconditions

All VCDUs in the Major\_Frame\_VCDU\_Set have the same VCID, same contact id, increasing mnf counters, and no data break. The set starts and ends with a pair of VCDUs having the mnf counter drop as the VCDU counter increases.

Postconditions

Output time code minor frames.

If the major frame synchronization is not found, the set of VCDUs in the Major\_Frame\_VCDU\_Set are placed into Failed\_Mjf\_data and the processing of the Major\_Frame\_VCDU\_Set ends.

If the end of line code is not found, then the set of VCDUs in the Major\_Frame\_VCDU\_Set are placed into Failed\_Mjf\_data and the processing of the Major\_Frame\_VCDU\_Set ends.

Constraints

None

Functional Breakdown

Search, from the beginning, each each minor frame in a Major\_Frame\_VCDU\_Set for the major frame synchronization code.

Place all VCDUs in the Major\_Frame\_VCDU\_Set into Failed\_Mjf\_data if the major frame synchronization is not found and end processing of the Major\_Frame\_VCDU\_Set

Search each minor frame of the Major\_Frame\_VCDU, starting from the expected location of the end of line code based on the location of the major frame synchronization code, for the end of line code.

Place all VCDUs in the Major\_Frame\_VCDU\_Set into Failed\_Mjf\_data if the end of line code is not found and end processing of the Major\_Frame\_VCDU\_Set

Extract the Time\_Code minor frames from the Major\_Frame\_VCDU\_Set.

Output the Time\_Code minor frames.

Reusability

The code for searching major frame synchronization, and end of line codes reusable for searching other bit patterns.

NAME:

3.3.2

TITLE:

Extract Major Frame Time

INPUT/OUTPUT:

Time\_Code : data\_in

Mjf\_Tossed\_Cnt : data\_out

Mjf\_Time\_Code\_Err\_Cnt : data\_out

Max\_Time\_Span : data\_in

Actual\_Stop\_Time : data\_in

Major\_Frame\_Time : data\_out

Time\_Range\_Tol : data\_in

Maj\_Vote\_Tol : data\_in

Mjf\_Data\_Rate : data\_in

BODY:

Description of Process

Extract and validate Major\_Frame\_Time from the six time code minor frames.

Assumptions

Preconditions

The six time code minor frames are available.

Postconditions

Major\_Frame\_Time is output.

Mjf\_Time\_Code\_Err\_Cnt is output.

Constraints

None.

Functional Breakdown

Validate the six time code minor frames.

Extract the Major\_Frame\_Time from the validated six time code minor frames.

Validate the Major\_Frame\_Time against the Actual\_Stop\_Time (from the RDC\_ACCT) and the Actual\_Start\_Time (calculated from the Actual\_Stop\_Time and the Max\_Time\_Span of the Valid\_MFP\_Parms).

Validate the Major\_Frame\_Time against the Major\_Frame\_Time of the previous major frame.

Estimate the major frame time incase the extracted major frame time is anomalous.

Output a flag to indicate valid/invalid status of the extracted Major\_Frame\_Time.

Output Major\_Frame\_Time

Reusability

The code used to validate the time code minor frames may be reusable.

NAME:

3.3.3

TITLE:

Collect VCDU Quality and Accounting

INPUT/OUTPUT:

VCDU\_QA : data\_inout

Major\_Frame\_VCDU\_Set : data\_in

Sub\_Intv\_Info : data\_in

Valid\_CCSDS\_Parms : data\_in

BODY:

Description of Process

Collect the VCDU quality and accounting data on a major frame basis.

Assumptions

Preconditions

Quality annotations exist in the Major\_Frame\_VCDU\_Set.Ann\_VCDU.

Postconditions

The Mjf\_VCDU\_QA store will be updated with the newly calculated information.

Constraints

None.

Functional Breakdown

Accumulate quality and accounting information into VCDU\_QA for each VCDU in the Major\_Frame\_VCDU\_Set.

Output VCDU\_QA.

Accumulate quality and accounting information into VCDU\_QA for each VCDU in the Failed\_Mjf\_Data.

Output VCDU\_QA.

Reusability

None.

NAME:

3.3.4

TITLE:

Determine Subintervals

INPUT/OUTPUT:

Mjf\_Full\_Fill\_Cnt : data\_inout

Current\_Sub\_Intv\_Id : data\_inout

Sub\_Intv\_Id : data\_out

Mjf\_Full\_Fill\_Cnt : data\_out

Sub\_Intv : data\_out

Major\_Frame\_Time : data\_in

Sub\_Intv\_Delta : data\_in

Sub\_Intv\_Info : data\_in

Mjf\_Data\_Rate : data\_in

MF\_Stop\_Time : data\_in

BODY:

Description of Process

Determine the subinterval range.

Assumptions

Preconditions

The Valid\_MFP\_Parms.Sub\_Intv\_Delta must exist.

A valid Major frame time value must exist.

Postconditions

A Subinterval ID and range are generated.

Constraints

None.

The start of a new subinterval is identified by one of the following conditions:

1. The VCID\_Change\_Flag of the Sub\_Intv\_Info is set to "true".
2. The delta time between the Major\_Frame\_Time and the MF\_Stop\_Time from Sub\_Intv exceeds the Sub\_Intv\_Delta of the Valid\_MFP\_Parms.
3. A backward jump in time.

Functional Breakdown

Check to see if a new subinterval is beginning.

If a new subinterval is beginning, then end the previous subinterval if any.

Update the Sub\_Intv store and Sub\_Intv data flow.

Calculate the number of missing major frames before the latest major frame being processed and place the result into the Mjf\_Full\_Fill\_Cnt.

Output the Mjf\_Full\_Fill\_Cnt.

Accumulate the Mjf\_Full\_Fill\_Cnt into the MFP\_Acct datastore.

Reusability

None.

NAME:

3.4.1

TITLE:

Deinterleave and Reverse Bands

INPUT/OUTPUT:

Aligned\_Bands : data\_in

Mjf\_VCDU\_Data : data\_in

Scan\_Dir : data\_in

BODY:

Description of Process

Extract and deinterleave band data for each detector from the minor frames of the Major\_Frame\_VCDU\_Set according to Scan\_Dir and format into Deinterleaved\_Band\_Data on a major frame basis

Assumptions

Preconditions

Major\_Frame\_VCDU\_Set has been validated and locations of major frame sync and end of line codes are known.

Postconditions

The Deinterleaved\_Band\_Data is organized by band and detector on a major frame basis.

Constraints

None.

Functional Breakdown

Deinterleave band data for each detector from the minor frames of the Major\_Frame\_VCDU\_Set into the Aligned\_Bands using forward or reverse Scan\_Dir and format type from Status\_Info.

Output Deinterleaved\_Band\_Data

Reusability

None

NAME:

3.4.2

TITLE:

Align Bands

INPUT/OUTPUT:

Major\_Frame\_Time : data\_in

Scan\_Dir : data\_in

Status\_Info : data\_in

Aligned\_Bands : data\_out

Mjf\_Full\_Fill\_Cnt : data\_in

Fill\_Value : data\_in

Sensor\_Alignment\_Info : data\_in

Sub\_Intv\_Id : data\_in

BODY:

Description of Process

Align the bands according to Sensor\_Alignment\_Info, Scan\_Dir, and format on a major frame basis.

Assumptions

Preconditions

The sensor alignment info is available.

Postconditions

The detectors of the Aligned\_Bands are aligned on a major frame basis.

Constraints

None

Functional Breakdown

For each missing major frame, (Mjf\_Full\_Fill\_Cnt in number),

Move the Sub\_Intv\_Id, the format type, major frame time, and end of contact values into Aligned\_Bands.

The major frame time may be interpolated.

Perform alignment for each ETM+ band using the Scan\_Dir, format type, and Sensor\_Alignment\_Info. The Scan\_Dir may be interpolated.

Output the Aligned\_Bands.

For the Major\_Frame\_VCDU\_Set, move the Sub\_Intv\_Id, the format type, major frame time, and end of contact values into Aligned\_Bands.

Perform alignment for each ETM+ band using the Scan\_Dir, format type, and Sensor\_Alignment\_Info.

Output the Aligned\_Bands.

Reusability

None

NAME:

3.5.1

TITLE:

Create MSCD File

INPUT/OUTPUT:

MSCD\_Data : data\_inout

Sub\_Intv\_Id : data\_out

MSCD\_File\_Name : data\_out

MFP\_MSCD\_Status : data\_out

MSCD\_File : data\_out

MF\_Start\_Time : data\_in

File\_Version\_Number : data\_in

Sub\_Intv\_Id : data\_in

BODY:

Description of Process

Creates the MSCD\_File\_Name for the MSCD\_File for each subinterval.

Stores MSCD\_Data in the MSCD\_File for each subinterval.

Assumptions

Preconditions

The Sub\_Intv\_Id is received.

Postconditions

The MSCD\_File\_Name and MSCD\_File is created for each subinterval.

Empties the MSCD\_Data store after each subinterval.

Send MFP\_Status.MFP\_MSCD\_Status to MACS.

The MSCD\_File\_Name is placed into the MFP\_Acct store associated with a particular subinterval

Constraints

None.

Functional Breakdown

Create the MSCD\_File\_Name using File\_Version\_Number and MF\_Start\_Time identified by Sub\_Intv\_Id.

Write the MSCD\_Data to the MSCD\_File.

Update the LPS\_File\_Info store to contain the MSCD\_File\_Name for the subinterval identified by Sub\_Intv\_Id.

Reusability

This process may be used to create the Cal\_File.

NAME:

3.5.2

TITLE:

Create Calibration File

INPUT/OUTPUT:

Cal\_Data : data\_inout

Sub\_Intv\_Id : data\_out

Cal\_File\_Name : data\_out

MFP\_Cal\_Status : data\_out

Cal\_File : data\_out

MF\_Start\_Time : data\_in

Sub\_Intv\_Id : data\_in

File\_Version\_Number : data\_in

BODY:

Description of Process

Creates the Cal\_File\_Name and the Cal\_File for each subinterval.

Stores the Cal\_Data in the Cal\_File for each subinterval.

Description of Process

Creates the Cal\_File\_Name and the Cal\_File for each subinterval.

Stores the Cal\_Data in the Cal\_File for each subinterval.

Assumptions

Preconditions

The Sub\_Intv\_Id is received.

Postconditions

The Cal\_File\_Name and the Cal\_File is created for each subinterval.

Emptys the Cal\_Data store after each subinterval.

Send MFP\_Cal\_Status to MACS.

The Cal\_File\_Name is placed into the MFP\_Acct store associated with a particular subinterval

Constraints

None.

Functional Breakdown

Create the Cal\_File\_Name using File\_Version\_Number and MF\_Start\_Time identified by Sub\_Intv\_Id.

Receive the Cal\_Data and write the data to the Cal\_File.

Update the LPS\_File\_Info store to contain the Cal\_File\_Name for the subinterval identified by Sub\_Intv\_Id.

Reusability

This process may be used to create the MSCD\_File.

NAME:

3.5.3

TITLE:

Extract MSCD Data

INPUT/OUTPUT:

Scan\_Dir : data\_out

MSCD\_Data : data\_out

Mjf\_VCDU\_Data : data\_in

Major\_Frame\_Time : data\_in

BODY:

Description of Process

Extracts the MSCD\_Data from the Major\_Frame\_VCDU\_Set and store the MSCD\_Data in the MSCD\_Data store.

Assumptions

Preconditions

The Major\_Frame\_VCDU\_Set is received.

Postconditions

The MSCD\_Data are extracted from the Major\_Frame\_VCDU\_Set for a major frame.

The MSCD\_Data is stored in the MSCD\_Data store.

Constraints

None

Functional Breakdown

Extract the two minor frames after the End\_Of\_Line of the Mjf\_VCDU\_Data.

Extract the SHS\_Err, FHS\_Err, and the Scan\_Dir from the two minor frames.

Append the Scan\_Dir, FHS\_Err, SHS\_Err and the Major\_Frame\_Time to the MSCD\_Data store

Output the Scan\_Dir.

Create and store fill data in MSCD\_Data store for each missing major frame.

Reusability

Use the majority vote of data words from the parse major frame process to extract the MSCD\_Data.

NAME:

3.5.4

TITLE:

Extract Calibration Data

INPUT/OUTPUT:

Cal\_Data : data\_out

Mjf\_VCDU\_Data : data\_in

Fill\_Value : data\_in

Sensor\_Alignment\_Info : data\_in

Major\_Frame\_Time : data\_in

Status\_Info : data\_in

Scan\_Dir : data\_in

BODY:

Description of Process

Extracts the Cal\_Data from the Major\_Frame\_VCDU\_Set for each subinterval and store the same in the CAL\_Data store.

Description of Process

Extracts the Cal\_Data from the Major\_Frame\_VCDU\_Set for each subinterval and store the same in the CAL\_Data store.

Assumptions

Preconditions

The Major\_Frame\_VCDU\_Set is received.

Postconditions

The deinterleaved Cal\_Data is accumulated in the Cal\_Data store for a subinterval.

Constraints

None.

Functional Breakdown

Search for the end of mirror scan correction data in Mjf\_VCDU\_Data.

Extract the Cal\_Data on a minor frame basis from Mjf\_VCDU\_Data.

Create and store fill data in Cal\_Data store for each missing major frame.

Align Cal\_Data using the Scan\_Dir, format type from Status\_info, and the Sensor\_Alignment\_info.

Deinterleave the Cal\_Data into the the aligned bands using the Scan\_Dir, format type from Status\_info, and the Sensor\_Alignment\_info.

Append the Major\_Frame\_Time to the Cal\_Data for every major frame.

Append the Status\_Info to the Cal\_Data for every major frame.

Append the aligned and deinterleaved Cal\_Data to the Cal\_Data store according to band width and detector number.

Reusability

Uses portions of the Deinterleave and Reverse Bands and Align Bands procedure from the Generate Band Data process.

NAME:

3.6

TITLE:

Collect Quality and Accounting

INPUT/OUTPUT:

Mjf\_VCDU\_QA : data\_inout

Mjf\_VCDU\_QA : data\_out

MFP\_Mjf\_Status : data\_out

Sub\_Intv\_Id : data\_out

Valid\_MFP\_Thres : data\_in

Sub\_Intv\_Id : data\_in

BODY:

Description of Process

Collect the Mjf\_VCDU\_QA data, check threshold values, and place the information into the accounting store on a subinterval basis.

Assumptions

Preconditions

VCDU\_QA has quality and accounting data on a major frame basis.

This process is invoked for each major frame.

Postconditions

The VCDU\_QA is accumulated into the MFP\_Acct store on a subinterval basis.

Constraints

None.

Functional Breakdown

Store the major frame quality and accounting information in Mjf\_VCDU\_QA.

Accumulate the major frame quality and accounting information, VCDU\_QA, for the entire subinterval.

Compare the subinterval quality and accounting information values against the threshold values of the Valid\_MFP\_Thres store, and output information messages when thresholds are exceeded.

Place validated Mjf\_VCDU\_QA values into the MFP\_Acct store.

Store the subinterval quality and accounting information in Mfp\_acct.

Empty the VCDU\_QA of all data pertaining to the major frame.

Reset the subinterval quality and accounting information values when a new subinterval begins.

Reusability

None.

---

## 4.6.2 Performance Requirements

The following performance requirements are allocated to the MFPS:

- 4.6.2.1 The MFPS software on each LPS string shall provide the capability to process the equivalent of any combination of the format 1 and format 2 portions of 125 Landsat 7 ETM+ scenes of wideband data per day (approximately 25 to 30 gigabytes per day).
- 4.6.2.2 The MFPS software on each LPS string shall process received data at a minimum rate of not less than 7.5 Mbps. (based on a minimum raw wideband throughput of 7.5 Mbps).
- 4.6.2.3 The MFPS software on each LPS string shall provide the capability to execute concurrently with all other LPS subsystems.
  - 4.6.2.3.1 The MFPS software shall begin to process received raw data immediately on receipt of required input.
  - 4.6.2.3.2 The MFPS software shall output the equivalent of one Landsat 7 ETM+ scene worth's of major frames and PCD within 240 seconds of the receipt of all required input.
- 4.6.2.4 The MFPS software on each LPS string shall provide the capability to reprocess a maximum of 10 percent of a string's daily input volume of wideband data (approximately 12.5 scenes or 2.5 to 3 gigabytes per day).
- 4.6.2.5 The MFPS software on each LPS string shall provide the capability to process received wideband data at a daily average aggregate rate of 3 Mbps (includes 10 percent of overhead due to reprocessing).
- 4.6.2.6 The MFPS software on each LPS string shall maintain data processing throughput performance for all Landsat 7 raw wideband data received with a BER of one bit error in  $10^5$  bits, without loss of LOR processed data and without retransmission.
- 4.6.2.7 The MFPS software on each LPS string shall output any periodic processing status information that it generates with a maximum latency of 30 seconds between output.

This page intentionally left blank.

---

## 4.7 Payload Correction Data Subsystem

The PCDS constructs PCD major frames out of PCD words received from the MFPS and creates output files containing those major frames for each subinterval in a contact. The PCDS also identifies WRS scenes for each subinterval and provides scene metadata to the MACS and the IDPS. The PCDS also extracts band presence information from the PCD and forwards it to the MACS for inclusion in the subinterval's metadata.

---

### 4.7.1 Functional Requirements

The following functional requirements are allocated to the PCDS:

- Synchronize on PCD bytes for assembling PCD minor frames.
- Assemble PCD major frames.
- Fill missing PCD data.
- Generate PCD files on a subinterval basis.
- Collect and store PCD quality and accounting on both subinterval and PCD major frame bases.
- Extract and store last instrument on and off times for each subinterval.
- Identify scenes in the WRS contained within each subinterval by path and row and compute scene metadata, including actual center and corner positions, Sun azimuth and elevation angle, and horizontal display shift.
- Automatically generate alarms to alert the operator to errors exceeding selected thresholds.

#### 4.7.1.1 Major Functions

The PCDS accepts pcd\_Info from the MFPS consisting of sequences of the four PCD bytes contained in each Landsat 7 VCDU. Each set of PCD bytes is accompanied by the subinterval sequence identifier of the subinterval to

which it belongs and with information describing gaps in the VCDU sequence.

For each subinterval, the PCDS locates sequences of unpacked PCD cycles and extracts a single, packed PCD word from each cycle. The PCDS assembles PCD minor frames from the resulting sequences of packed PCD words and then constructs PCD major frames from the minor frames. The PCDS assigns a time tag to each PCD major frame based on the timecodes appearing in instances of PCD major frame 0.

The PCDS accumulates quality and accounting information during frame construction and summarizes this information on both a subinterval and a major frame basis. The PCDS also extracts band presence information from each occurrence of PCD major frame 2 for which the time is known. The PCDS stores both kinds of information for later use by the MACS.

The PCDS extracts and validates the spacecraft position from each PCD major frame. Using the spacecraft position, the PCDS identifies the time at which the spacecraft passed over each scene center within a subinterval. For each such scene center, the PCDS computes scene metadata and forwards it to the IDPS and the MACS.

For each subinterval, the PCDS creates an output file and writes the PCD major frames to the file. As part of this process, the PCDS converts angular display sensor (ADS) samples, ADS temperature, gyro, and gyro drift samples to engineering units.

The major functions of PCDS are depicted in Figures 4-11 through 4-14.

#### **4.7.1.2 Detailed Functional Requirements**

This section contains the process specifications for the lowest level processes in the PCDS data flow diagrams.

NAME:

4.1.1

TITLE:

Locate PCD Sequences

INPUT/OUTPUT:

PCD\_Info : data\_in

unpackd\_PCD\_sequences : data\_out

missing\_word\_counts,unpackd\_word\_counts : data\_out

missing\_word\_counts : data\_out









**BODY:****Description of Process**

Collect PCD into sequences by subinterval

**Assumptions****Preconditions**

None.

**Post conditions**

None.

**Constraints**

None.

**Functional Breakdown**

Output a set of sequences of unpacked PCD words for each subinterval in the contact with each set of PCD sequences accompanied by the subinterval's subinterval sequence ID. Each sequence is the longest possible sequence of the input unpacked PCD words in pcd\_Bytes such that the associated lps\_missingVCDUCnt and lps\_dataBrk values are 0 and LPS\_FALSE, respectively, for all but the first pcd\_Bytes.

For each sequence of unpacked PCD words, output a corresponding value of missing\_word\_counts as follows. If the sequence is followed by another sequence in the same subinterval, then output the value of lps\_missingVCDUCnt associated with the first pcd\_Bytes of the next sequence. Otherwise, output 0.

For each sequence of unpacked PCD words, output a corresponding value of unpacked\_word\_counts equal to the number of words in the sequence.

**Reusability**

None.

NAME:

4.1.2

TITLE:

Sync on Unpacked PCD

INPUT/OUTPUT:

unpacked\_PCD\_sequences : data\_in

unpacked\_PCD\_words : data\_out

BODY:

Description of Process

Locate unpacked PCD cycles in the stream of unpacked PCD words.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

None.

Functional Breakdown

For each subinterval in the contact, output the subinterval sequence ID and a series of unpacked PCD word sequences derived from the input unpacked PCD sequences as follows (cf. Specification Document 3, 3.2.7.1).

For each input sequence, locate each occurrence of the pattern "0x16, B1, B2, B3 0x32+" (B1, B2, and B3 are any byte values) and output an unpacked data word set consisting of B1, B2, and B3.

The last occurrence of the pattern in a sequence may be truncated to "0x16 B1 B2." In this case, output B1 and B2 alone.

Reusability

None.

NAME:

4.1.3

TITLE:

Pack PCD

INPUT/OUTPUT:

unpacked\_PCD\_words : data\_in

packed\_PCD\_words,vote\_results : data\_out

vote\_results : data\_out

BODY:

Description of Process

Produce a packed PCD word by bitwise majority vote.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

None.

Functional Breakdown

For each subinterval in the contact, output the subinterval's subinterval sequence ID and a sequence of packed PCD words derived from the input unpacked PCD words by majority voting the words in the set as follows. If the unpacked data word set contains exactly 2 words, then the result of majority voting is the first data word. Otherwise, the i-th bit (i = 0 to 7) of the resulting packed PCD word is the i-th bit of the first data word, if the i-th bits of the first and second data words are equal. Otherwise, the i-th bit of the result is the value of the i-th bit of the third data word.

For each unpacked data word set in each sequence of PCD words in each subinterval of the contact, output a vote\_results value as follows. If there are three words in the data word set and all three are equal, then the vote\_result is SUCCESS. Otherwise, the vote\_result is FAILURE.

Reusability

None.

NAME:

4.1.4

TITLE:

Assemble PCD Minor Frames

INPUT/OUTPUT:

packed\_PCD\_words, vote\_results : data\_in

pcd\_minor\_frames, pcd\_minor\_vote\_result\_counts, pcd\_minor\_sync\_errors\_counts :  
data\_out

BODY:

Description of Process

Construct PCD minor frames from sequences of packed PCD

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

None.

Functional Breakdown

For each sequence of packed PCD words in each subinterval in the contact, output the subinterval's subinterval sequence ID and a sequence of minor frames constructed from the packed PCD words in the sequence as follows. Locate and output up to 125 bytes following the first occurrence of the PCD minor frame sync pattern specified in Specification Document 3, Table 10-2, as the first PCD minor frame in the sequence. Output fewer than 125 bytes for the first minor frame if and only if the sequence contains fewer than 125 bytes after the sync pattern.

For each series of 128 bytes remaining in the sequence, output the last 125 bytes. Output the remaining bytes at the end of the sequence as a partial minor frame, if and only if the number of bytes remaining is greater than 3.

For each minor frame generated, output LPS\_TRUE as the value of pcd\_minor\_frame\_sync\_errors, if the value of the first three bytes of the 128 byte sequence making up the minor frame equal the minor frame sync pattern. Otherwise, output LPS\_FALSE.

For each minor frame generated, output a pcd\_minor\_frame\_vote\_counts value equal to the number of vote\_results values equal to FAILURE associated with the minor frame's constituent PCD words.

Reusability

Renaissance Frame Sync Building Block to perform minor frame synchronization.

NAME:

4.1.5

TITLE:

Assemble PCD Major Frames

INPUT/OUTPUT:

pcd\_major\_frame\_ID\_errors, pcd\_major\_frame\_unpacked\_words\_count, pcd\_major\_frame\_missing\_words\_count, pcd\_major\_frame\_vote\_err\_count, pcd\_major\_frame\_sync\_err\_counts : data\_out

missing\_word\_counts, unpacked\_word\_counts : data\_in

pcd\_minor\_frames, pcd\_minor\_vote\_result\_counts, pcd\_minor\_sync\_errors\_counts : data\_in

pcd\_major\_frames : data\_out

BODY:

Description of Process

Construct PCD major frames from sequences of PCD minor frames

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

None.

Functional Breakdown

For each sequence of PCD minor frames in each subinterval in the contact, output the subinterval's subinterval sequence ID and major frames consisting of successive sequences of PCD minor frames such that the minor frame ID counter ascends for each successive PCD minor frame and the major frame is terminated when minor frame ID 127 is encountered.

For each major frame, assign a major frame ID as follows. If word 72 of minor frame 96 is present (cf. Specification Document 3, 3.2.7.4.10 and Table 12) and its most significant three bits are all 1, then assign the frame ID 0. Otherwise, if word 72 from any of minor frames 96 through 103 are present, then the major frame ID is whichever of the values 1, 2, or 3 appears in the majority of those word 72 instances (cf. Specification Document 3, 3.2.7.4.14 and Table 14). Otherwise, the major frame ID is 0.

For each major frame generated, output quality information as follows.

Output a pcd\_major\_frame\_vote\_err\_count value equal to the sum of the pcd\_minor\_frame\_vote\_result\_count values associated with the major frame's constituent minor frames. Output a pcd\_major\_frame\_sync\_err\_counts value equal to the number of pcd\_minor\_frame\_sync\_errors\_counts values equal to LPS\_TRUE associated with its constituent minor frames. Output a pcd\_major\_frame\_ID\_errors value equal to LPS\_TRUE, if the frame's ID is not the next expected. Otherwise, output LPS\_FALSE. Output a pcd\_major\_frame\_unpacked\_words\_count value equal to the sum of the unpacked\_word\_counts associated with the major frame's constituent minor frames.

Reusability  
None.

NAME:

4.1.6

TITLE:

Assign PCD Major Frame Times

INPUT/OUTPUT:

time\_tagged\_pcd\_major\_frames : data\_out

pcd\_major\_frames : data\_in

BODY:

Description of Process

Compute time tags for each PCD major frame

Assumptions

Preconditions

None.

Post conditions

None.

Constraints

None.

Functional Breakdown

For each major frame in each subinterval in the contact, output the subinterval's subinterval sequence ID and the major frame with a time tag computed as follows.

If the major frame ID (as specified in Specification Document 3, 3.2.7.4.14) is 0, then the time tag of the major frame is the value of the time code included in the frame itself (cf. Specification Document 3, 3.2.7.4.10). However, if any component of the time code is outside its defined range, then the time tag is the value of the time of the previous major frame plus 4.096 seconds (cf. Specification Document 3, 3.2.7.2). If the previous major frame in the sequence does not have a time tag or if a previous major frame does not exist, then the timetag for the major frame is NULL.

If the major frame ID is not 0 and there is an instance of major frame 0 previous to the frame in question with a non-null time tag, then the time tag of the major frame is the value of the nearest previous instance of major frame 0 plus 4.096 seconds multiplied by the major frame's ID. If there is no previous instance of major frame 0 with a non-NULL time tag, then the time tag for the frame in is NULL.

Reusability

None.

NAME:

4.1.7

TITLE:

Issue Missing Words Alarm

INPUT/OUTPUT:

missing\_word\_counts : data\_in

num\_missing\_data\_words : data\_in

pcd\_missing\_words\_message : data\_out

BODY:

Description of Process

Output alarm message if number of missing words exceeds an operator-set threshold.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

None.

Functional Breakdown

If the sum of missing\_word\_counts is greater than num\_missing\_data\_words, then output pcd\_missing\_words\_message to notify the operator of the number of missing words.

Reusability

None.

NAME:

4.1.8

TITLE:

Issue Majority Vote Failure Alarm

INPUT/OUTPUT:

num\_failed\_votes : data\_in

vote\_results : data\_in

pcd\_failed\_votes\_message : data\_out

BODY:

Description of Process

Output alarm message if the number of majority vote failures exceeds an operator-set threshold.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

None.

Functional Breakdown

If the sum of the instances of vote\_results equal to FAILURE is greater than num\_failed\_votes, then output pcd\_failed\_votes\_message to notify the operator of the number of majority vote failures.

Reusability

None.

NAME:

4.2

TITLE:

Summarize PCD Quality

INPUT/OUTPUT:

PCD\_Mjf\_Acct : data\_out

PCD\_Acct : data\_out

time\_tagged\_PCD\_major\_frames, pcd\_major\_frame\_vote\_err\_counts, pcd\_major\_frame\_sync\_err\_counts : data\_in

spacecraft\_position\_quality : data\_in

UT1\_corrections : data\_in

BODY:

Description of Process

Aggregate PCD quality and accounting on major frame and subinterval bases.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

None.

Functional Breakdown

For each subinterval in the contact, output a record to pcd\_Acct as follows.

The Sub\_Intv\_Sequence\_Id is the subinterval sequence ID for the subinterval. The First\_PCD\_Mjf\_Time is equal to the time tag of the first major frame in the subinterval. The PCD\_Stop\_Time is equal to the time tag of the last major frame in the subinterval. The ETM\_Last\_On\_Time is the extracted from the last instance of major frame 0 in the subinterval in which word 72 of minor frames 42-47 are present (cf. Specification Document 3, 3.2.7.4.16 and Table 17). The ETM\_Last\_Off\_Time is the value extracted from the last instance of major frame 0 in the subinterval in which word 72 of minor frames 84-89 are present (cf. Specification Document 3, 3.2.7.4.16 and Table 17).

For each major frame with a non-null time tag in each subinterval in the contact, output a record to PCD\_Mjf\_Acct as described below. Include a record for each missing major frame as well, where missing major frames are inferred from gaps in the sequence of time tags.

Each record is output as follows. The Sub\_Intv\_Sequence\_Id is the subinterval sequence ID of the subinterval. The PCD\_Mjf\_Time is the time tag of the major frame. The PCD\_Words\_Received is the number of words in the major frame. The Failed\_PCD\_Votes is the value of pcd\_major\_frame\_vote\_err\_counts. The Total\_PCD\_MNF is 128. The Num\_PCD\_MNF\_Sync\_Errors is the value of pcd\_major\_frame\_sync\_err\_counts. The Num\_PCD\_Filled\_MNF is equal to the number of minor frames missing at least one word. The PCD\_Filled\_MJF\_Flag is 'E', if the major frame is missing, 'P' if major frame is missing at least one word, and 'N', if neither of the above are true. The PCD\_ADP\_MJF\_Flag is 'G', if the value of spacecraft\_position\_quality

associated with the attitude extracted from the major frame is VALID, is 'R', if the value of spacecraft\_position\_quality is INVALID, and is 'M', if the value of spacecraft\_position\_quality is MISSING. The PCD\_EDP\_Mjf\_Flag is 'G', if the value of the spacecraft\_position\_quality associated with the ephemeris extracted from the major frame is VALID, is 'R', if the value of spacecraft\_position\_quality is INVALID, and is 'M', if the value of spacecraft\_position\_quality is MISSING.

Reusability  
None.

NAME:

4.3

TITLE:

Extract Bands Present

INPUT/OUTPUT:

Bands\_Present : data\_out

time\_tagged\_PCD\_major\_frames : data\_in

BODY:

Description of Process

Extract band presence from PCD and forward to the MACS

Assumptions

Preconditions

None.

Post conditions

None.

Constraints

None.

Functional Breakdown

For each occurrence of major frame 2 with a non-NULL time tag and in which word 72 of minor frames 32 and 35 are present, output a record to Bands\_Present as follows.

The Sub\_Intv\_Sequence\_Id is the subinterval sequence ID of the subinterval. The PCD\_Cycle\_Time is the major frame's time tag. The bits of Band\_Present are derived from word 72 of minor frames 32 and 35 as follows (where bit 0 = the least significant and bit 7 = the most significant bit).

Band_Present Bit	Source Bit
0	Minor frame 32, word 72, bit 1
1	Minor frame 32, word 72, bit 2
2	Minor frame 32, word 72, bit 3
3	Minor frame 32, word 72, bit 4
4	Minor frame 32, word 72, bit 5
5	Minor frame 32, word 72, bit 6
6	Minor frame 32, word 72, bit 7
7	Minor frame 35, word 72, bit 7

Cf. Specification Document 3, Table 19.

Reusability

None.

NAME:

4.4

TITLE:

Extract Spacecraft Position

INPUT/OUTPUT:

spacecraft\_positions : data\_out

time\_tagged\_PCD\_major\_frames : data\_i

BODY:

Description of Process

Extract attitude and ephemeris points from PCD

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

None.

Functional Breakdown

For each major frame in each subinterval of the contact in which either word 72 of minor frames 0 through 15 are all present or word 72 of minor frames 50 through 73 all are present, output a spacecraft\_position value with a time tag equal to the major frame time tag minus 8.192 seconds, an attitude point extracted from word 72 of minor frames 0 through 15 (cf. Specification Document 3, 3.2.4.5) or NULL, if any of those minor frames are missing, and an ephemeris point extracted from word 72 of minor frames 50 through 73 (cf. Specification Document 3, 3.2.4.8) or NULL, if any of those minor frames are missing.

Reusability

None.

NAME:

4.5

TITLE:

Validate Spacecraft Position

INPUT/OUTPUT:

spacecraft\_position\_quality : data\_out

spacecraft\_positions : data\_in

valid\_ephemeris\_range, valid\_ephemeris\_cross\_product\_range, attitude\_quaternion\_tolerance : data\_in

BODY:

Description of Process

Verify that attitude and ephemeris are within tolerance.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

None.

Functional Breakdown

For each element in spacecraft\_positions, output MISSING as the value of the attitude quality element of spacecraft\_position\_quality for this point, if the attitude is missing, VALID, if the sum of squares of the elements of the attitude equals 1.0 plus or minus attitude\_quaternion\_tolerance, and INVALID in all other cases Output MISSING as the value of the ephemeris quality element of spacecraft\_position\_quality for this point, if the ephemeris is missing, VALID if each ephemeris velocity component falls within the upper and lower bounds for ephemeris velocity in valid\_ephemeris\_range, each ephemeris position component falls within the upper and lower bounds for ephemeris position in valid\_ephemeris\_range, and the cross product of the position and velocity components falls within the range defined by ephemeris\_cross\_product\_range, and INVALID in all other cases.

Reusability

None.

NAME:

4.6.1

TITLE:

Compute Look Points

INPUT/OUTPUT:

look\_points : data\_out

spacecraft\_positions, spacecraft\_position\_quality : data\_in

Valid\_Scene\_Parms : data\_in

BODY:

Description of Process

Compute the latitude and longitude of the ground point viewed at scan center for each spacecraft position.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

Use the algorithm specified in Specification Document 10

Functional Breakdown

For each valid element of spacecraft\_positions (as indicated by spacecraft\_position\_quality), use the algorithm specified in Specification Document 10 to output the time and look point latitude and longitude.

Reusability

General purpose functions from the RSL library.

NAME:

4.6.2

TITLE:

Locate Scene Centers

INPUT/OUTPUT:

UT1\_Corrections : data\_out

scene\_center\_times : data\_out

scene\_center\_actual\_positions : data\_out

scene\_center\_time\_and\_position : data\_out

look\_points : data\_in

UTC\_UT1\_Corrections : data\_in

nominal\_center\_latitudes : data\_in

BODY:

Description of Process

Identify the time at which each scene center latitude was crossed.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

Use the algorithm specified in Specification Document 10).

Functional Breakdown

For each nominal center latitude in Valid\_WRS\_Parms that is bracketed by latitudes from look\_points (indicating that the center latitude was crossed between the times of the two look points), output the time at which the scene center latitude was crossed and the actual latitude and longitude of the look point at center scan nearest the nominal scene center.

Output the same information for any scene center that was crossed such that the scene center crossing time was prior to the time of the first look\_point or after the time of the last look point of the subinterval and for which any portion of the scene is present in the subinterval.

For each subinterval, output the UT1 correction used to compute the longitude.

Reusability

General purpose functions from the RSL library.

NAME:

4.6.3

TITLE:

Compute Corner Positions

INPUT/OUTPUT:

corner\_positions : data\_out

corner\_times\_and\_positions : data\_out

scene\_center\_time\_and\_position : data\_in

MF\_Start\_Time, MF\_Stop\_Time : data\_in

BODY:

Description of Process

Compute the latitudes and longitudes of the four corners of each scene identified.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

Use the algorithm specified in Specification Document 10

Functional Breakdown

For each scene center identified, use the algorithm specified in Specification Document 10) to compute the latitude and longitude of the actual scene corners. If the first scene in a subinterval begins before the start of image data for the subinterval (i.e., when the scene center time minus 1/2 the duration of the time to cover an entire scene is less than the MF\_Start\_Time for the subinterval), then the starting scene corners are the corners of the first scan in the subinterval. If the last scene in the subinterval ends after the end of the subinterval (i.e. when the scene center time plus 1/2 the duration of the time to cover an entire scene is greater than the MF\_Stop\_Time for the subinterval), then the ending corners of the scene are the corners of the last scan in the subinterval.

Reusability

None.

NAME

4.6.4

TITLE:

Compute HDS

INPUT/OUTPUT:

HDS : data\_out

scene\_center\_actual\_positions : data\_in

nominal\_center\_positions,previous\_center\_positions : data\_in

BODY:

Description of Process

Compute the Horizontal Display Shift (HDS) for each scene center.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

Use the algorithm specified in Specification Document 10

Functional Breakdown

For each scene center in each subinterval, use the algorithm specified in Specification Document 10 to compute the HDS.

Reusability

None.

NAME:

4.6.5

TITLE:

Compute Sun Position

INPUT/OUTPUT:

sun\_positions : data\_out

sun\_elevation : data\_out

scene\_center\_time\_and\_position : data\_in

BODY:

Description of Process

Compute the sun azimuth and elevation at each identified scene center.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

Use the algorithm specified in Specification Document 10

Functional Breakdown

For each identified scene center, use the algorithm specified in Specification Document 10 to output the sun azimuth and elevation angle.

Reusability

None.

NAME:

4.6.6

TITLE:

Assemble Scene Messages

INPUT/OUTPUT:

scene\_Info : data\_out

corner\_times\_and\_positions : data\_in

sun\_elevation : data\_in

paths\_and\_rows : data\_in

scene\_center\_time\_and\_position : data\_in

BODY:

Description of Process

Send messages to the IDPS describing each scene identified.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

None.

Functional Breakdown

For each scene identified in each subinterval, output a scene\_info message with type SCENE\_INFO\_MESSAGE and containing the scene center time, scene start time (earlier corner time), scene stop time (later corner time), center and corner latitude/longitudes, path and row number, and sun elevation. Output the scenes messages in order of ascending scene center time. After the output of the sequence of scene messages for a subinterval, output a scene\_info message with type END\_OF\_SUBINTERVAL\_MSG. If the subinterval is the last in the contact, then output LPS\_TRUE as the value of the message's End\_Of\_Contact\_Flag. Otherwise output LPS\_FALSE as the value of the message's End\_Of\_Contact\_Flag.

Reusability

None.

NAME:

4.6.7

TITLE:

Assemble Scene Metadata

INPUT/OUTPUT:

PCD\_Scene\_Acct : data\_out

HDS : data\_in

paths\_and\_rows : data\_in

scene\_center\_time\_and\_position : data\_in

corner\_positions : data\_in

sun\_positions : data\_in

cal\_door\_activity\_for\_scenes : data\_in

BODY:

Description of Process

Store scene metadata for later retrieval by the MACS.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

None.

Functional Breakdown

For each scene identified in each subinterval of the contact, insert a record into PCD\_Scene\_Acct containing the subinterval sequence ID, the scene center time, center and corner latitude/longitudes, path and row, Horizontal Display Shift (HDS), sun position (azimuth and elevation angle), and calibration door activity status.

Reusability

None.

NAME:

4.6.8

TITLE:

Report Cal Door Activity

INPUT/OUTPUT:

cal\_door\_activity\_for\_scenes : data\_out

scene\_center\_times : data\_in

time\_tagged\_PCD\_major\_frames : data\_in

BODY:

Description of Process

Report calibration door activity status for each identified scene.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

None.

Functional Breakdown

For each identified scene in each subinterval, output a value for

cal\_door\_activity\_for\_scenes as follows:

If the instance of serial word P of Major Frame 2 (cf. Specification Document 3, Table 19) with the greatest time tag value less than the scene center time (or the least greatest time tag, if no instance of Major Frame 2 has a lesser time tag) has either bit 2 or 3 set, then output the value 1. Otherwise, output 0.

Reusability

None.

NAME:

4.7.1

TITLE:

Convert ADS to Engineering Units

INPUT/OUTPUT:

ADS\_in\_EUs : data\_out

time\_tagged\_PCD\_major\_frames : data\_in

BODY:

Description of Process

Convert Angular Displacement Sensor (ADS) samples extracted from PCD into engineering units.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

Cf. conversion specification in Specification Document 9, 4.1.3.3.1.

Functional Breakdown

For each ADS sample in each major frame (cf. Specification Document 3, 3.2.7.4.1 and Table 10-1) in each subinterval of the contact, output the sample converted to engineering units according to the extraction and conversion specification in Specification Document 9, 4.1.3.3.1.

Reusability

None.

NAME:

4.7.2

TITLE:

Convert ADS Temperature to Engineering Units

INPUT/OUTPUT:

ADS\_Temperatures\_in\_EUs : data\_out

time\_tagged\_PCD\_major\_frames : data\_in

BODY:

Description of Process

Convert Angular Displacement Sensor (ADS) temperature samples extracted from PCD into engineering units.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

Cf. conversion specification in Specification Document 9, 4.1.3.3.2.

Functional Breakdown

For each ADS temperature sample in each major frame (cf. Specification Document 3, 3.2.7.4.2 and Tables 17 through 20) in each subinterval of the contact, output the sample converted to engineering units according to the extraction and conversion specification in Specification Document 9, 4.1.3.3.2.

Reusability

None

NAME:

4.7.3

TITLE:

Convert Gyro to Engineering Units

INPUT/OUTPUT:

Gyro\_in\_EUs : data\_out

time\_tagged\_PCD\_major\_frames : data\_in

BODY:

Description of Process

Convert Gyro pulse counts extracted from PCD to engineering units

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

Cf. conversion specification in Specification Document TBD, 4.1.3.3.3.

Functional Breakdown

For each Gyro pulse count in each major frame (cf. Specification Document 3, 3.2.7.4.3) in each subinterval of the contact, output the pulse count converted to engineering units according to the extraction and conversion specification in Specification Document 9, 4.1.3.3.3.

Reusability

None.

NAME:

4.7.4

TITLE:

Convert Gyro Drift to Engineering Units

INPUT/OUTPUT:

Gyro\_Drift\_in\_EUs : data\_out

time\_tagged\_PCD\_major\_frames : data\_in

BODY:

Description of Process

Convert Gyro drift samples extracted from PCD into engineering units.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

Cf. conversion specification in Specification Document 9, 4.1.3.3.4.

Functional Breakdown

For each Gyro drift sample in each major frame (cf. Specification Document 3, 3.2.7.4.4 and Table 17) in each subinterval of the contact, output the sample converted to engineering units according to the extraction and conversion specification in Specification Document 9, 4.1.3.3.4.

Reusability

None.

NAME:

4.7.5

TITLE:

Create PCD Output Files

INPUT/OUTPUT:

PCD\_Files : data\_out

ADS\_in\_EUs : data\_in

ADS\_Temperatures\_in\_EUs : data\_in

Gyro\_in\_EUs : data\_in

Gyro\_Drift\_in\_EUs : data\_in

time\_tagged\_PCD\_major\_frames,pcd\_major\_frame\_ID\_errors,pcd\_major\_frame\_unpacked\_words\_counts,pcd\_major\_frame\_missing\_words\_counts,pcd\_major\_frame\_vote\_err\_counts,pcd\_major\_frame\_sync\_err\_counts,spacecraft\_positions,spacecraft\_position\_quality : data\_in

pcd\_file\_names : data\_in

BODY:

Description of Process

Write PCD for each subinterval to a separate HDF file.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

Cf. Specification Document 9, Table 4-10.

Functional Breakdown

For each subinterval in the contact, create a PCD output file containing a single vdata with name equal to the value of pcd\_file\_name for the subinterval.

For each major frame in each subinterval, output a vdata record according to the specification in Specification Document 9, Table 4-10. Output fill values for missing fields as specified.

Reusability

Use COTS HDF library to create and write files.

NAME:

4.7.6

TITLE:

Create PCD File Names

INPUT/OUTPUT:

pcd\_file\_names : data\_out

LPS\_Hardware\_String\_Id,Capture\_Source,Actual\_Start\_Time : data\_in

file\_version\_number,subinterval\_sequence\_IDs : data\_in

VCID,Contact\_Sequence\_ID : data\_in

BODY:

Description of Process

Construct the name for each PCD output file.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

Cf. Specification Document 9, 3.4.

Functional Breakdown

For each subinterval in the contact, construct a file name as follows.

Retrieve the subinterval's VCID (Virtual Channel ID) and  
Contact\_Sequence\_ID from sub\_Intv using the  
subinterval\_sequence\_ID as key.

Retrieve the LPS\_Hardware\_String\_Id, Capture\_Source, and  
Actual\_Start\_Time from rdc\_acct using the Contact\_Sequence\_Id as  
key.

Use the capture source to compute the X band.

Use the VCID as the format number.

Use the year, day, and hour of the Actual\_Start\_Time as the start time of  
the capture.

Use the ordinal position of the subinterval within the sequence of  
subintervals in the contact as the subinterval number.

Use the file\_version\_number as the fileversion number.

Reusability

None.

---

## 4.7.2 Performance Requirements

The following performance requirements are allocated to the PCDS:

- 4.7.2.1 The PCDS software on each LPS string shall provide the capability to process the equivalent of any combination of the format 1 and format 2 portions of 125 Landsat 7 ETM+ scenes of wideband data per day (approximately 25 to 30 gigabytes per day).
- 4.7.2.2 The PCDS software on each LPS string shall process unpacked PCD data at a minimum rate of not less than 3.2 kilobits per second (kbps) (based on a minimum raw wideband throughput of 7.5 Mbps).
- 4.7.2.3 The PCDS software on each LPS string shall provide the capability to execute concurrently with all other LPS subsystems.
- 4.7.2.3.1 The PCDS software shall begin to process received raw wideband data immediately on receipt of required input.
- 4.7.2.3.2 The PCDS software shall output a scene center identification, a Sun azimuth at scene center value, and a Sun elevation at scene center value within 240 seconds of the time of receiving all required input.
- 4.7.2.4 The PCDS software on each LPS string shall provide the capability to reprocess a maximum of 10 percent of a string's daily input volume of wideband data (approximately 12.5 scenes or 2.5 to 3 gigabytes per day).
- 4.7.2.5 The PCDS software on each LPS string shall provide the capability to process unpacked PCD data at a daily average aggregate rate of 12.7 kbps per second (includes 10 percent of overhead due to reprocessing).
- 4.7.2.6 The PCDS software on each LPS string shall maintain data processing throughput performance for all Landsat 7 raw wideband data received with a BER of one bit error in  $10^5$  bits, without loss of level-zero processed data and without retransmission.

- 4.7.2.7 The PCDS software on each LPS string shall output any periodic processing status information that it generates with a maximum latency of 30 seconds between output.

---

## **4.8 Image Data Processing Subsystem**

The IDPS is responsible for producing band and browse image files, generating cloud coverage assessment scores, and displaying scan data as it is being processed in a moving window.

### **4.8.1 Functional Requirements**

The following functional requirements are allocated to the IDPS:

- Generate multiband browse data for each ETM+ image on a subinterval basis for format 1 data.
- Generate a band file for each band received on a subinterval basis.
- Perform automatic cloud cover assessment (ACCA) for WRS scenes using predefined comparison values on scene quadrant and full scene basis for format 1 data.
- Produce an MWD that displays scan data as it is received by the IDPS.

#### **4.8.1.1 Major Functions**

In its Band File Generation process, the IDPS takes aligned band data from the MFPS, separates it by subinterval and band, and produces band files. In the Browse File Generation process, the aligned band data is separated by scene and then reduced by a waveletting algorithm to produce multiband browse files. In the ACCA process, the aligned band data is separated by scenes, and then an ACCA algorithm is used on the data to produce ACCA scores. A moving window display displays scans as they are processed. Finally, accounting information is produced for metadata generation and the band and browse files are made available to the LDTS.

The major functions of the IDPS are depicted in Figures 4-15 through 4-17.







#### 4.8.1.2 Detailed Functional Requirements

This section contains the process specifications for the lowest level processes in the IDPS DFDs.

NAME:

5.1

TITLE:

IDPS Main

INPUT/OUTPUT:

IDP\_Browse\_Status : data\_in

IDP\_ACCA\_Status : data\_in

IDP\_Band\_Status : data\_in

IDPS\_Status : data\_out

BODY:

Description of Process

This module is the parent process of the whole IDP subsystem. It starts the Generate Moving Window Display, the Generate Band File, the Generate Browse File, and the Perform ACCA processing. It also monitors the completion status of each of its children. If one of the children fails, it will terminate the others and return a status to MACS.

Assumptions

Preconditions

A shared memory segment is available with the MFPS.

Postconditions

The IDPS child processes have completed their processing.

Constraints

None.

Functional Breakdown

Create a named pipe to be shared between the Generate Band File module and the Generate Moving Window Display module.

Start the following functions as child processes:

Generate Band File

Generate Moving Window Display

Generate Browse File

Perform ACCA

Wait for each child process to complete.

Receive IDP\_Band\_Status from Generate Band File

Receive IDP\_Browse\_Status from Generate Browse File

Receive IDP\_ACCA\_Status from Perform ACCA

\*NOTE: The Moving Window Display does not return status to this function because its processing is handled by the X-Windows server.

If IDP\_Band\_Status, IDP\_Browse\_Status, and IDP\_ACCA\_Status contain success conditions

Set IDPS\_Status to success

Otherwise

Set IDPS\_Status to a failure condition  
Remove the named pipe and any remaining child processes.  
Return IDPS\_Status to MACS.  
Terminate.

Reusability  
None.

NAME:

5.2

TITLE:

Generate Band File

INPUT/OUTPUT:

Sun\_Elevation : data\_out

End\_Of\_Subinterval : data\_out

End\_Of\_Contact : data\_out

Gain\_States : data\_out

Band\_File\_Names : data\_out

IDP\_Band\_Status : data\_out

File\_Name : data\_out

Scan\_Data : data\_out

Band\_Acct : data\_out

Aligned\_Bands : data\_in

Scene\_Info : data\_in

BODY:

Description of Process

This function generates one file for each band on a subinterval basis. The files contain scan data in the form of an HDF-EOS swath and geolocation data. The files are generated using the HDF-EOS API.

Assumptions

Preconditions

None.

Postconditions

Three or six band files have been generated.

Band\_Acct has been passed to the IDP\_Acct datastore.

Constraints

None.

Functional Breakdown

Receive each major frame of Aligned\_Bands data from MFP shared memory.

Extract the Scan\_Data from the Aligned\_Bands.

Send a copy of the Scan\_Data to the Moving Window Display (idp\_mwd).

Extract the Band\_Gains from the Aligned\_Bands.

If format 1 data, then

Write the Scan\_Data for bands 1-6 in six separate files, one file for each band.

If format 2 data, then

Write the Scan\_Data for bands 6, 7, and Pan in three separate files, one for each band.

If end of subinterval

Send an End\_Of\_Subinterval notification to the MWD.

Receive Scene\_Info scene definition information from PCD.

Extract the scene start, center, and stop times from the PCD Scene\_Info.

Extract the Sun\_Elevation from the PCD Scene\_Info.

Determine the scene boundaries in the Aligned\_Bands data by comparing the PCD Scene\_Info start, center and stop times against the times in

the MFP Aligned\_Bands data and finding the major frames which most closely match them.  
Write these scene boundaries to the Band\_Files.  
Write the scene corner and center latitudes and longitudes to the Band\_Files.  
Close the band files.  
If format 1 data  
    Send Band\_File\_Names to the Generate Browse File module.  
    Send Band\_File\_Names to the Perform ACCA module.  
    Calculate the Band\_Acct data from the Scene\_Info and Aligned\_Bands data.  
    Place the Band\_Acct in the IDP\_Acct datastore.  
    Place Aligned\_Bands.Sub\_Intv\_Id into Band\_Acct.Sub\_Intv\_Id.  
    Place the Band\_File\_Names into LPS\_FILE\_INFO.  
    Extract Gain\_States from the Aligned\_Bands data.  
    Place Gain\_States into Band\_Gain\_States.  
    Calculate the Gain\_Change\_Flag.  
    Place Gain\_Change\_Flag and Gain\_Change\_Scan\_Number into Band\_Gain\_States.  
    Send Band\_File\_Names to the Generate Browse File module.  
    Send Band\_File\_Names to the Perform ACCA module.  
    Send the Sun\_Elevation to the Perform ACCA module.  
If end of contact  
    Send End\_Of\_Contact to the Generate Browse File module.  
    Send End\_Of\_Contact to the Perform ACCA module.  
    Send End\_Of\_Contact to the MWD.  
    Return a success condition in IDP\_Band\_Status to IDPS Main.  
    Terminate.  
If any of the above operations cannot be performed  
    Log an error  
    Return a failure condition in IDP\_Band\_Status to IDPS Main.  
    Terminate.  
If recoverable errors occur in the above processing  
    Log an error.

Reusability  
None.

NAME:

5.3.1

TITLE:

Correct and Contrast Stretch Image

INPUT/OUTPUT:

JPEG\_Quality : data\_out

Wavelet\_Iterations : data\_out

bitplain : data\_out

Band\_Filenames : data\_in

Scan\_Data : data\_in

End\_Of\_Contact : data\_in

Gains\_and\_Biases : data\_in

Rad\_Corr\_Parms : data\_in

BODY:

Description of Process

This function performs radiometric correction and contrast stretching on a browse image.

Assumptions

Preconditions

There is a contrast stretching factor,

Valid\_Band\_Parms.Contrast\_Stretch\_Factor is stored in

Valid\_Band\_Parms. Band files containing aligned band data on a subinterval basis have been created from which image data can be extracted on a scene by scene basis.

Postconditions

A radiometrically corrected and contrast stretched image has been produced in the form of a bitplain which can be passed to other modules for further processing.

Constraints

This processing is performed only for format 1 data and only on bands 1 through 5.

Functional Breakdown

Get Valid\_Band\_Parms.Subs from datastore Valid\_Band\_Parm.

For each new subinterval, extract the following band/color associations from datastore Val\_Band\_Parms:

Valid\_Band\_Parms.Multi1 -- red

Valid\_Band\_Parms.Multi2 -- green

Valid\_Band\_Parms.Multi3 -- blue

Get the Band\_Filenames from an interprocess message sent by the Generate Band File module.

Get the radiometric correction parameters Rad\_Corr\_Parms from a file or database.

Get the Gains\_and\_Biases from Valid\_Detector\_Gain\_Bias.

Extract the contrast stretch factor from Valid\_Band\_Parms.

Extract the JPEG\_Quality factor from Valid\_Band\_Parms.

Extract the Wavelet\_Iterations from Valid\_Band\_Parms.

For each scene in each of the three selected band files,

Extract the Scan\_Data by scene from the band file.

Perform radiometric correction on each pixel in the scene using  
Gains\_and\_Biases and Rad\_Corr\_Parms.  
Perform contrast stretching on each pixel in the scene using the contrast  
stretch factor.  
Pass bitplain containing the adjusted scene, the Wavelet\_Iterations, and  
the JPEG\_Quality to module Reduce Image by Subsamples.

#### Reusability

The radiometric correction function is shared with the Perform ACCA module.  
The function which reads the band files is shared with the Perform ACCA module

NAME:

5.3.2

TITLE:

Reduce Image by Wavelets

INPUT/OUTPUT:

IDP\_Browse\_Status : data\_out

Browse\_Filenames : data\_out

Image : data\_out

bitplain : data\_in

Wavelet\_Iterations : data\_in

JPEG\_Quality : data\_in

BODY:

Description of Process

This function reduces an image by replacing a grid of pixels with one pixel where the intensity of that one pixel is derived from a set of adjacent pixels using a waveletting algorithm. The image is that of one scene of data and is written to an HDF file on a scene basis.

Assumptions

Preconditions

A function to perform wavelet reduction is available.

Postconditions

Multi band browse files have been created containing wavelet reduced and JPEG compressed RIS24 images produced from three predetermined bands. The browse images are written as HDF files on a scene basis. LPS\_File\_Info contains an entry for the browse file.

Constraints

Browse files will not be created for format 2 data.

Functional Breakdown

Reduce bitplain by passing it to Wavelet\_Iterations of the wavelt\_alg function obtained for the LPS project.

Perform JPEG compression on the reduced image to the level of JPEG\_Quality.

Create names for the browse files in accordance with LPS standards,

Browse\_Filenames.

Write the image to an HDF-format file using the HDF DF24 API.

Place Browse\_Filenames in the LPS\_File\_Info table.

Set IDP\_Browse\_Status to success if no errors occurred. Otherwise, set it to a failure condition.

Reusability

No LPS produced software can be reused. However, the wavelet\_alg function obtained for LPS and the NCSA HDF library will be used.

NAME:

5.4.1

TITLE:

Collect Scene Data

INPUT/OUTPUT:

Sun\_Elevation : data\_out

Scene : data\_out

Band\_File\_Names : data\_in

Scan\_Data : data\_in

Rad\_Corr\_Parms : data\_in

Gains\_and\_Biases : data\_in

End\_Of\_Contact : data\_in

Sun\_Elevation : data\_in

BODY:

Description of Process

This function extracts a scene of scan data from the band files containing data from bands 2 through 6. It performs radiometric correction on each pixel and then passes the scene to the Generate Cloud Cover Assessment module.

Assumptions

Preconditions

Band files must exist and contain scene information.

Postconditions

Five ACCA scores will be generated for the scene, one for each quadrant and one overall score. The ACCA scores will be placed in the LPS database.

Constraints

Cloud Cover Assessment is only performed on Format 1 data. At least one scene's worth of data must be available to be collected.

Functional Breakdown

Get the Rad\_Corr\_Parms from the Radiometric\_Correction\_Parms datastore.

Get the Gains\_and\_Biases from the Valid\_Detector\_Gain\_Bias datastore.

Get the number of scenes in the subinterval from a band file.

For each identified scene

For each of the files (bands 2-6) in Band\_File\_Names

Extract a line of Scan\_Data and buffer it.

Apply radiometric correction to each pixel using Rad\_Corr\_Parms and Gains\_and\_Biases.

Internally buffer the Scan\_Data for each band as a scene.

Pass the Scene to the Generate Cloud Cover Assessment module.

Pass the Sun\_Elevation to the Generate Cloud Cover Assessment module.

If any of the above operations fail

Log an error

Terminate.

Reusability

The radiometric correction function is reused with the Generate Browse File module.

The function which extracts data from the band files is reused with the Generate Browse File module.

NAME:

5.4.2

TITLE:

Generate Cloud Cover Assessment

INPUT/OUTPUT:

ACCA\_Scores : data\_out

IDP\_ACCA\_Status : data\_out

Scene : data\_in

Sun\_Elevation : data\_in

BODY:

Description of Process

The Generate Cloud Cover Assessment function determines the percentage of cloud coverage on a scene quadrant and full scene basis.

Assumptions

Preconditions

None.

Postconditions

Five percentage scores have been generated and put into the IDP\_ACCT table for each scene: one for each of the four quadrants and an aggregate score for the whole scene. Status messages are sent indicating that ACCA has been performed on a specific scene.

Constraints

ACCA is generated only on format 1 data.

Functional Breakdown

Find the start of each quadrant in Scene.

Use Sun\_Elevation to calculate a threshold used by the assessment algorithm.

For each quadrant in the scene

Examine data values against the cloud cover assessment thresholds and algorithm.

Compile the cloud cover percentage quadrant scores.

Average the quadrant assessments for an aggregate scene score.

Place the quadrant and aggregate scores in ACCA\_Scores.

Place ACCA\_Scores into the IDP\_Acct store for this scene.

If processing completed normally

Set IDP\_ACCA\_Status to a success condition.

Else

Set IDP\_ACCA\_Status to a failure condition.

Log an error.

Reusability

The algorithm used for this function was patterned after a prototype that was provided. Because the prototype was not written in C, it was necessary to rewrite it. However, the logic from the prototype was preserved.

NAME:

5.5

TITLE:

Generate Moving Window Display

INPUT/OUTPUT:

End\_Of\_Subinterval : data\_in

Display : data\_out

Valid\_MWD\_Parms : data\_in

End\_Of\_Contact : data\_in

Scan\_Data : data\_in

BODY:

Description of Process

This module receives major frames from the Generate Band File module, converts them into displayable pixel lines, reduces the lines, and displays them as an image on a display terminal as they are received. The Moving Window Display (MWD) displays the image in a window and scrolls it as more major frames are received. It also displays the current major frame time, the data format (1 or 2), the subinterval sequence ID, and the band/color associations. The MWD concurrently with band file generation. The MWD processes both format 1 and format 2 data.

Assumptions

Preconditions

The X-Windows/Motif environment must be available on the system and an X-Windows compatible color terminal or workstation must be available to display the image. The rest of the IDPS must be up and running.

Postconditions

A viewable image of the swath is displayed on the screen and disappears at the end of the contact.

Constraints

None.

Functional Breakdown

Spawn two MWD processes.

Establish communication between the parent and child process using a pipe.

Parent MWD process:

Open communication with the Generate Band File module via a named pipe.

Receive messages through the named pipe from the Generate Band File module

Case: the received message contains Scan\_Data

Forward the message to the MWD child process through the pipe.

Case: the received message contains an End\_Of\_Contact notification

Notify the child process.

Terminate all MWD processing.

Case: the received message contains and End\_Of\_Subinterval notification

Notify the child process.

Child MWD process:

Register the file descriptor of the parent-child pipe as an X-Windows event.

When a message is received handle the event as follows:

Case: The message contains Scan\_Data

Reduce the size of the display data so that it will fit into a window by pixel averaging.

Map the pixel values to an X-Windows color palette.

Output the pixels to Display.

Update the labels on Display.

Case: The message contains and End\_Of\_Subinterval notification

Output a ten-pixel high horizontal bar of black pixels to display demarcate a subinterval boundary.

Case: The message contains an End\_Of\_Contact notification

Delete Display.

Terminate processing.

#### Reusability

Some code developed for an X-Windows MWD prototype was reused.

---

## 4.8.2 Performance Requirements

The following performance requirements are allocated to the IDPS:

- 4.8.1 The IDPS software on each LPS string shall provide the capability to process the equivalent of any combination of the format 1 and format 2 portions of 125 Landsat 7 ETM+ scenes of wideband data per day (approximately 25 to 30 gigabytes per day).
- 4.8.2 The IDPS software on each LPS string shall process aligned band data at a minimum rate of not less than 7.5 Mbps (based on a minimum raw wideband throughput of 7.5 Mbps without PCD and CADU overhead).
- 4.8.3 The IDPS software on each LPS string shall provide the capability to execute concurrently with all other LPS subsystems.
- 4.8.3.1 The IDPS software on each LPS string shall begin to process received data immediately on receipt of required input.
- 4.8.3.2 The IDPS software on each LPS string shall output scene metadata within 250 seconds of the time of receiving all required input.
- 4.8.4 The IDPS software on each LPS string shall provide the capability to reprocess a maximum of 10 percent of a string's daily input volume of data (approximately 12.5 scenes or 2.5 to 3 gigabytes per day).
- 4.8.5 The IDPS software on each LPS string shall provide the capability to process received data at a daily average aggregate rate of 2.9 Mbps (includes 10 percent of overhead due to reprocessing).
- 4.8.6 The IDPS software on each LPS string shall maintain data processing throughput performance for all Landsat 7 raw data received with a BER of one bit error in  $10^5$  bits, without loss of level-zero processed data and without retransmission.
- 4.8.7 The IDPS software on each LPS string shall output any periodic processing status information that it generates with a maximum latency of 30 seconds between output.

This page intentionally left blank.

---

## 4.9 Management and Control Subsystem

The MACS is responsible for system-level control and monitoring of LPS devices and processes. It also provides the interface between the LPS and the operator and generates metadata files on a subinterval basis.

---

### 4.9.1 Functional Requirements

The following functional requirements are allocated to the MACS:

- Provide an orderly system startup and shutdown capability.
- Provide the capability to monitor and control LPS operation, including the display of error messages or alarms to the operator.
- Generate LOR metadata files on a subinterval basis, using information received from other subsystems.
- Provide the capability to generate, display, and/or print LPS summary or quality and accounting information on operator request Data Receive Summary Report, LPS Quality Assurance Report, and LPS File Transfer Summary Report).
- Provide monitoring test points and indicators to verify proper operation of system capabilities and components.
- Provide the capability to manually enable or disable LPS automated functions or to suspend LPS file generation.
- Configure system resources and modify parameter setup tables and error reporting threshold tables.
- Ingest contact schedules received from the Mission Operations Center (MOC).
- Ingest calibration parameter files received from the Image Assessment System (IAS).

#### 4.9.1.1 Major Functions

On receipt of an operator-keyed directive, the MACS first determines which subsystem the directive belongs to and forwards it to the target subsystem.

If the Directive is MAC\_Ingest\_Schedule\_Drct, MAC\_Ingest\_Cal\_File\_Drct, or MAC\_Modify\_LOR\_Parms\_Thres\_Drct, it sends the directive to the Ingest Contact Schedule, Ingest Calibration File, or Modify LOR Parameters and Threshold function to modify the appropriate datastore. If the directive is MAC\_Control\_Drct, it sends it to the LPS System Control function for LPS system-level control.

When Meta\_Data\_Drct is issued, the Contact\_Id is forwarded to the Generate Metadata process, which generates a metadata file for each subinterval in the contact period.

During LPS operation, the MACS monitors for system failure. It receives the processing status (RDP\_Return\_Status, MFP\_Return\_Status, PCD\_Return\_Status, or IDP\_Return\_Status) from the subsystems and displays it to the operator. It also reports the quality and accounting information and summary information (Data Receive Summary Report, LPS Quality Assurance Report, and Data Transfer Summary Report) on operator request.

The major functions of the MACS are depicted in Figures 4-18 through 4-20.

#### 4.9.1.2 Detailed Functional Requirements

This section contains the process specifications for the lowest level processes in the MACS DFDs.

NAME:

6.1

TITLE:

Process LPS Directive

INPUT/OUTPUT:

RDC\_Drct : data\_out

Report\_Drct : data\_out

MAC\_Gen\_Metadata\_Drct : data\_out

MAC\_Drct\_Dispatch\_Status : data\_out

MAC\_Ingest\_Schedule\_Drct : data\_out

MAC\_Modify\_LOR\_Parms\_Thres\_Drct : data\_out

MAC\_Control\_Drct : data\_out

Directive : data\_in

**BODY:**

**Description of Process**

Accept the Directive from LPS operator and send the Directive to the target subsystem(s) or the MACS internal functions.







## Assumptions

## Preconditions

None.

## Postconditions

All Directives are sent to their appropriate destination.

## Constraints

None.

## Functional Breakdown

Accept the Directive keyed from LPS operator.

Forward the Directive to the appropriate subsystem(s) as shown in the following:

RDC\_Directive to RDCS

RDP\_Directive to RDPS

MFP\_Directive to MFPS

PCD\_Directive to PCDS

IDP\_Directive to IDPS

LDT\_Directive to LDTS

MAC\_Ingest\_Cal\_Drct to Ingest Cal File Function

MAC\_Ingest\_Schedule\_Drct to Ingest Contact Schedule Function

MAC\_Modify\_LOR\_Parms\_and\_Thres\_Drct to Modify LOR Parms and Thres  
Function

MAC\_Control\_Drct to LPS System Control

When MAC\_Drct\_Metadata\_Drct is issued,

Send Contact\_Id to the Generate Metadata process.

Send the directive dispatching status as MAC\_Directive\_Dispatch\_Status to  
Report LPS Status function for display.

## Reusability

None.

NAME:

6.2

TITLE:

Generate Metadata

INPUT/OUTPUT:

Metadata\_File : data\_out

MAC\_Gen\_Metadata\_Drct : data\_in

Contact\_Id : data\_in

BODY:

Description of Process

Create the Metadata\_File using the quality and accounting information on a subinterval basis upon completion of LOR processing.

Assumption

Preconditions

The latitude and longitude information for corner quadrants is available from Valid\_WRS\_Parms.

Postconditions

The Metadata\_File is created on a subinterval basis.

Constraints

None.

Functional Breakdown

Receive the Contact\_Id (Contact\_Sequence\_Id and File\_Version\_Number).

For each subinterval associated with the current Contact\_Id, do the following:

Verify that all of the quality and accounting files are completed for the current processed Sub\_Intv.Sub\_Intv\_Id.

Create the Metadata\_File\_Name

Create the Metadata\_Header with the following information:

Current\_Time

LPS\_Configuration

Generate the subinterval information from the following tables:

Bands\_Present

Band\_Gain\_States

LPS\_File\_Info

MFP\_Acct

MFP\_MJF\_Acct

PCD\_Acct

PCD\_MJF\_Acct

Sub\_Intv

For each scene associated with the current subinterval, generate the scene information from the following tables:

IDP\_Acct

PCD\_Scene\_Acct

Reusability

None.

NAME:

6.3

TITLE:

Report LPS Status

INPUT/OUTPUT:

LPS\_Journal : data\_out

MAC\_Drct\_Dispatch\_Status : data\_in

LDT\_Status : data\_in

MAC\_Control\_Status1 : data\_in

MAC\_Ingest\_Cal\_Status : data\_in

MAC\_Ingest\_Schedule\_Status : data\_in

BODY:

Description of Process

Report or display the LPS\_Status to LPS operator.

Assumption

Preconditions

None.

Postconditions

The LPS\_Status is displayed on the console to LPS operator.

Constraints

None.

Functional Breakdown

Receive incoming status messages from LPS Subsystems and other MACS functions.

Report the received status messages to the LPS operator as LPS\_Status.

Log the LPS\_Status message into the LPS\_Journal system log.

Reusability

The UNIX syslog and sysmon facilities are used to log and monitor the LPS.

NAME:

6.4

TITLE:

Ingest Cal File

INPUT/OUTPUT:

MAC\_Ingest\_Cal\_Drct : data\_in

BODY:

Description of Process

Modify and Ingest information in the Calibration Parameter file.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

None.

Functional Breakdown

Verify the MAC\_Ingest\_Cal\_Drct input and make sure the individual field format and range of the entry are valid.

Modify existing Calibration Parameter information.

Insert new Calibration Parameter information.

Ingest Calibration Parameters to multiple strings simultaneously.

Reusability

None.

NAME:

6.5

TITLE:

Modify LOR Parms and Thres

INPUT/OUTPUT:

MACS\_Modify\_Config\_Drct : data\_inout

MACS\_Modify\_Config\_Status : data\_out

MACS\_Modify\_Config\_Drct : data\_in

BODY:

Description of Process

Modify information in the LOR parameter and threshold stores.

Assumption

Preconditions

None.

Postconditions

None.

Constraints

None.

Functional Breakdown

Verify the MAC\_Modify\_LOR\_Parms\_Thres\_Drct input and make sure the individual field format and range of the entry are valid.

Modify entries in the LPS\_Configuration, UTC\_UT1\_Corrections, Valid\_Band\_Parms, Valid\_Detector\_Gain\_Bias, Valid\_WRS\_Parms, Valid\_CCSDS\_Parms, Valid\_LDT\_Parms, Valid\_MFP\_Parms, Valid\_Sensor\_Align\_Parms, Valid\_PCD\_Parms, Valid\_Scene\_Parms, Valid\_PCD\_Thres, Valid\_RDP\_Thres, and Valid\_MFP\_Thres stores.

Reusability

None.

NAME:

6.6

TITLE:

Ingest Contact Schedule

INPUT/OUTPUT:

Contact\_Schedules : data\_inout

MAC\_Ingest\_Schedule\_Status : data\_out

MAC\_Ingest\_Schedule\_Drct : data\_in

BODY:

Description of Process

Modify and Ingest information in the Contact\_Schedules file.

Assumption

Preconditions

None.

Pos conditions

None.

Constraints

None.

Functional Breakdown

Verify the MAC\_Ingest\_Schedule\_Drct input and make sure the individual field format and range of the entry are valid.

Modify existing Contact\_Schedules information.

Insert new Contact\_Schedules information.

Delete existing Contact\_Schedules information.

Ingest Contact\_Schedules information to multiple strings simultaneously.

Reusability

None.

NAME:

6.7

TITLE:

LPS System Control

INPUT/OUTPUT:

MAC\_Control\_Drct : data\_in

BODY:

Description of Process

Perform LPS control activity.

Assumption

Preconditions

None.

Postconditions

None.

Constraints

None.

Functional Breakdown

Bring up the LPS system if MAC\_Control\_Drct is Startup.

Shut down the LPS system if MAC\_Control\_Drct is Shutdown.

Start/Stop manual data capture if MAC\_Control\_Drct is Data Capture Start/Stop.

Start/Stop manual LOR data processing if MAC\_Control\_Drct is Data Processing

Start/Stop.

Start/Stop manual data copy to tape if Mac\_Control\_Drct is Data Copy to Tape

Start/Stop.

Start/Stop manual data copy to disk if Mac\_Control\_Drct is Data copy to Disk

Start/Stop.

Start/Stop automatic data capture if Mac\_Control\_Drct is Auto Data Capture

Start/Stop.

Generate tape label if Mac\_Control\_Drct is Generate Tape Label.

Assign MAC\_Control\_Status the result of any action taken.

Reusability

None.

NAME:

6.8

TITLE:

Monitor System Faults

INPUT/OUTPUT:

LPS\_Status : data\_out

LPS\_Journal : data\_in

BODY:

Description of Process

Allows the operator access to the Activity log for LPS.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

None.

Functional Breakdown

Display the contents of LPS\_Journal as LPS\_Status.

Reusability

The UNIX syslog and sysmon facilities are used to log and monitor the LPS.

NAME:

6.9.1

TITLE:

Validate RDP Parameters

INPUT/OUTPUT:

RDP\_Thresholds : data\_out

RDP\_Setup\_Status : data\_out

RDP\_CCSDS\_Parms : data\_out

RDP\_CCSDS\_Parms : data\_in

RDP\_Thresholds : data\_in

BODY:

Description of Process

Validate the CCSDS Parameters and RDP Thresholds received from the Operator and store the valid values.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

The parameters and thresholds must conform to predefined range values.

Functional Breakdown

Validate the Valid\_CCSDS\_Parms and Valid\_RDP\_Thres field values.

Invalid values entered by the operator will be rejected by Oracle Forms until correct values are entered.

Reusability

None.

NAME:

6.9.2

TITLE:

Validate MFP Parameters

INPUT/OUTPUT:

unnamed flow : data\_out

MFP\_Parms : data\_out

MFP\_Setup\_Status : data\_out

MFP\_Thresholds : data\_out

MFP\_Parms : data\_in

MFP\_Thresholds : data\_in

BODY:

Description of Process

Validate the MFP Parameters and MFP Thresholds received from the Operator and store the valid values.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

The parameters and thresholds must conform to predefined range values.

Functional Breakdown

Validate the Valid\_MFP\_Parms and Valid\_MFP\_Thres field values.

Invalid values entered by the operator will be rejected by Oracle Forms until correct values are entered.

Reusability

None.

NAME:

6.9.3

TITLE:

Validate IDP Parameters

INPUT/OUTPUT:

Valid\_MWD\_Parms : data\_out

Valid\_Detector\_Gain\_Bias : data\_out

Valid\_Band\_Parms : data\_out

IDP\_Setup\_Status : data\_out

IDP\_Band\_Parms : data\_in

BODY:

Description of Process

Validate the IDP Parameters received from the Operator and store the valid values.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

The parameters must conform to predefined range values.

Functional Breakdown

Validate the Valid\_MWD\_Parms, Valid\_Band\_Parms and Valid\_Detector\_Gain\_Bias field values.

Invalid values entered by the operator will be rejected by Oracle Forms until correct values are entered.

Reusability

None.

NAME:

6.9.4

TITLE:

Validate PCD Parameters

INPUT/OUTPUT:

Valid\_Scene\_Parms : data\_out

UTC\_UT1\_Corrections : data\_out

PCD\_Setup\_Status : data\_out

Valid\_PCD\_Thres : data\_out

Valid\_PCD\_Parms : data\_out

PCD\_Parms : data\_in

PCD\_Thresholds : data\_in

BODY:

Description of Process

Validate the PCD Parameters and PCD thresholds received from the Operator and store the valid values.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

The parameters must conform to predefined range values.

Functional Breakdown

Validate the Valid\_MWD\_Parms, Valid\_PCD\_Thres and UTC\_UT1\_Corrections field values.

Invalid values entered by the operator will be rejected by Oracle Forms until correct values are entered.

Reusability

None.

NAME:

6.9.5

TITLE:

Validate RDC LDT Parameters

INPUT/OUTPUT:

LPS\_Configuration : data\_out

RDC\_LDT\_Setup\_Status : data\_out

Valid\_LDT\_Parms : data\_out

RDC\_LDT\_Parms : data\_in

BODY:

Description of Process

Validate the LDT and RDC Parameters received from the Operator and store the valid values.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

The parameters must conform to predefined range values.

Functional Breakdown

Validate the Valid\_LDT\_Parms and LPS\_Configuration field values.

Invalid values entered by the operator will be rejected by Oracle Forms until correct values are entered.

Reusability

None.

NAME:

6.10.1

TITLE:

Generate Data Receive Summary Report

INPUT/OUTPUT:

Report\_RDC\_Data\_Capture\_Sum : data\_out

RDC\_Acct : data\_in

RDC\_Rpt\_Data\_Capture\_Sum\_Drct : data\_in

BODY:

Description of Process

This process generates the Data Receive Summary Report for a user-selected contact period.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

None.

Functional Breakdown

Select the following data from the RDC\_Acct table to be output to the report:

Capture string

Capture source

Raw data file name

Scheduled start time

Scheduled stop time

Actual start time

Actual stop time

Scheduled data volume

Expected data volume

Received data volumes

Transmission rate

Calculate the approximate number of Landsat scenes and output to the report:

(calculation: data volume (in megabytes)/number of megabytes per scene)

Scheduled scenes = scheduled data volume/224

Expected scenes = expected data volume/224

Received scenes = received data volume/224

Reusability

None.

NAME:

6.10.2

TITLE:

Generate Level OR QA Report

INPUT/OUTPUT:

Report\_LOR\_QA : data\_out

MFP\_Acct : data\_in

Rpt\_LOR\_QA\_Drct : data\_in

MF\_Start\_Time : data\_in

MF\_Stop\_Time : data\_in

RDC\_Acct : data\_in

LPS\_Configuration : data\_in

Valid\_CCSDS\_Parms : data\_in

RDP\_Acct : data\_in

IDP\_Acct : data\_in

MFP\_MJF\_Acct : data\_in

BODY:

Description of Process

This process generates the LPS Quality and Accounting Report for a user-selected contact period.

Assumptions

Preconditions

None

Postconditions

None

Constraints

None.

Functional Breakdown

Get the raw data info from the RDC\_Acct table:

Capture file name

Capture string

Contact start time

Contact stop time

Data volume

Get the tolerances from the Valid\_CCSDS\_Parms table:

Search

Check

Bit slip extent

Flywheel

Check error

Lock error

Using the RDP\_Acct, IDP\_Acct, MFP\_Acct, MFP\_MJF\_Acct, and the Sub\_Intv tables select/calculate the following information for each subinterval associated with the selected contact period:

Subinterval number

Subinterval start time

Subinterval stop time  
Number of CADUs  
Number of bytes of CADUs  
Number of scenes  
Number of major frames  
Number of partially filled major frames  
Number of entirely filled major frames  
Number of CADUs with sync errors  
Number of flywheel CADUs  
Number of missing CADUs  
Number of correctable VCDU headers (format 1)  
Number of correctable VCDU headers (format 2)  
Number of uncorrectable VCDU headers  
Number of CADUs with corrected BCH errors in the mission data zone  
Number of CADUs with corrected BCH errors in the data pointer zone  
Number of CADUs with uncorrected BCH errors in the mission data zone  
Number of CADUs with uncorrected BCH errors in the data pointer zone  
Bit error rate (sum of errors (bit)/10\*\*6 \* 9)

Reusability  
None.

NAME:

6.10.3

TITLE:

Generate File Transfer Summary Report

INPUT/OUTPUT:

Report\_LDT\_File\_Xfer\_Sum : data\_out

LDT\_Rpt\_File\_Xfer\_Sum\_Drct : data\_in

Current\_Time : data\_in

LPS\_File\_Info : data\_in

BODY:

Description of Process

This function generates the LPS file transfer summary report for a user-specified time period.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

None.

Functional Breakdown

Using the LPS\_File\_Info, RDC\_Acct, Processing\_Version\_Info, and the Sub\_Intv tables, select/calculate the following data to be output to the report:

Number of band files available

Number of PCD files available

Number of MSCD files available

Number of browse files available

Number of metadata files available

Number of calibration files available

Total number of files available

Number of band files transferred

Number of PCD files transferred

Number of MSCD files transferred

Number of browse files transferred

Number of metadata files transferred

Number of calibration files transferred

Total number of files transferred

Names of files available

Names of files transferred

Reusability

None.

---

## 4.9.2 Performance Requirements

The following performance requirements are allocated to the MACS:

- 4.9.2.1 The MACS software on each LPS string shall forward any directive to start or stop data capture or to generate a data receive summary to the RDCS within 1 second of its receipt from the operator.
- 4.9.2.2 The MACS software on each LPS string shall display a data receive summary for the most recently received raw wideband data within 1 second of its receipt from the RDCS.
- 4.9.2.3 The MACS software on each LPS string shall submit a data receive summary for the most recently received raw wideband data to a print queue within 1 second of its receipt from the RDCS.
- 4.9.2.4 The MACS software on each LPS string shall provide the capability to execute concurrently with all other LPS subsystems.
  - 4.9.2.4.1 The MACS software on each LPS string shall begin to process metadata immediately on receipt of required input.
  - 4.9.2.4.2 The MACS software on each LPS string shall output a metadata file within 240 seconds of the time of receiving all required input.
- 4.9.2.5 The MACS software on each LPS string shall maintain data processing throughput performance for all Landsat 7 raw wideband data received with a BER of one bit error in  $10^5$  bits, without loss of level-zero processed data and without retransmission.
- 4.9.2.6 The mean time to bring up the MACS software on any LPS string (from operating system boot to readiness to accept operator input) shall not exceed 12 minutes [based on a 15-minute estimate from reliability, maintainability, and availability (RMA) analysis and allowing 3 minutes for operator initiation and network latencies, 5 minutes for operating system boot, 5 minutes for DBMS startup, and 2 minutes for LPS software startup].
- 4.9.2.7 The time to bring up the MACS software on any LPS string (from operating system boot to readiness to accept operator input) shall

not exceed twice the required mean time to bring up the MACS software in 99 percent of all cases.

- 4.9.2.8 The MACS software on each LPS string shall output any periodic processing status information that it generates with a maximum latency of 30 seconds between output.

---

## **4.10 LPS Data Transfer Subsystem**

The LDTS is responsible for sending a DAN to the LP DAAC about the availability of LPS files on a contact basis. After receiving a data delivery notice (DDN), the LDTS is responsible for deleting the successfully transferred files. It is also responsible for generating a daily file transfer summary report on request.

---

### **4.10.1 Functional Requirements**

The following functional requirements are allocated to the LDTS:

- Interface with the LP DAAC to coordinate the transfer of LPS output files to the LP DAAC.
- Notify the LP DAAC of the availability of LPS files.
- Provide the capability to receive notification from LP DAAC on the successful archival of transferred LPS files.
- Provide the capability to store LPS data files until confirmation of successful transfer is received from the LP DAAC.
- Provide a manual override and protected capability to delete all LPS files on a specific contact period basis.
- Provide a manual override and protected capability to retain all LPS files online on a specific contact period basis.
- Provide the LPS file(s) transfer summary to the MACS for generating the file transfer summary report.
- Provide the capability to manually override the LPS automated functions.
- Provide the capability to selectively enable or disable the Transfer LPS Files function.

#### 4.10.1.1 Major Functions

The Generate\_DAN function receives the LDT\_Send\_DAN directive from the MACS and then retrieves information from the LDT\_File\_Info, LPS\_Configuration, and Sub\_Intv datastores to build a DAN.

The DAN is passed to the Send\_DAN function, which establishes a communication link between LPS and the LP DAAC and sends it to the LP DAAC. After a DAN is sent to LP DAAC, the Send\_DAN function waits for the DAA from LP DAAC. After receiving a DAA from the LP DAAC, the Send\_DAN functions verifies any DAN errors and stores the status information into the LDT\_File\_Trans\_Info datastore. The Send\_DAN function can be enabled and disabled by operator via the Control\_Send\_DAN function. The DAN can be resent on operator request (LDT\_Resend\_DAN\_Drct).

The Receive\_DDND function continuously receives and processes the DDND from the LP DAAC and stores the status of data transfer and archival at the LP DAAC into the LDT\_File\_Trans\_Info and LPS\_File\_Info datastores. After receipt of a DDND indicating successful transfer of a contact period file, the Receive\_DDND function notifies the Delete\_LPS\_Files function to delete the successfully transferred LPS output files. LPS output file deletion can be overridden by operator via the LDT\_Retain\_Files\_Drct function. The operator also can invoke the Delete\_LPS\_Files function to manually delete any contact period files if the files are not marked for retention.

The LDT\_File\_Trans\_Info datastore is created by the Generate\_DAN function and is maintained by the Generate\_DAN, Send\_DAN, Delete\_LPS\_Files, and Retain\_LPS\_Files functions. It is used by the MACS to build an LDTS Transfer Summary Report.

The major functions of the LDTS are depicted in Figure 4–21.

#### 4.10.1.2 Detailed Functional Requirements

This section contains the process specifications for the lowest level processes in the LDTS DFDs.

NAME:

7.1

TITLE:

Generate DAN



**INPUT/OUTPUT:**

LDT\_File\_Trans\_Info : data\_inout

DAN : data\_out

LDT\_Send\_DAN : data\_in

LPS\_File\_Info : data\_in

LPS\_Configuration : data\_in

Sub\_Intv\_List : data\_in

**BODY:****Description of Process**

This function generates a data availability notification (DAN) for a contact period, grouped by a subinterval, using LPS file names from the LPS\_File\_Info datastore, Sub\_Intv\_List from Sub\_intv datastore, LDTs processing parameters from Valid\_LDT\_Parms datastore, and LPS setup parameters from the LPS\_Configuration datastore (see constraints). Contact period file set and subinterval file group information needed by the Generate Transfer Summary function is stored in the LDT\_File\_Trans\_Info datastore.

**Assumptions****Preconditions**

None

**Postconditions**

None

**Constraints**

DAN format will be defined in the ICD between LPS and LP DAAC.

**Functional Breakdown**

This function shall retrieve from LPS\_File\_Info file characteristic such as file path, file name and file size for constructing the DAN.

This function shall retrieve LPS setup parameters (such as LPS\_Hardware\_string\_Id, LPS\_User\_Id, LPS\_Port\_Number, etc.) from the LPS\_Configuration datastore to be used to construct a DAN and establish the communication channel between LPS and LP DAAC.

This function shall retrieve the subinterval list for current contact period from Sub\_Intv datastore to be used as a file group basis.

This function shall build a DAN with the retrieved information for current contact period, group the subinterval LOR files as a file group aggregate

This function shall store LPS LOR file transfer accounting information based on a contact period and the subintervals into LDT\_File\_Trans\_Info datastore.

This function shall invoke "Send DAN" function to send the DAN over to LP DAAC.

**Reusability**

Some DDF and Pacor II DAN functionality was reused.

NAME:

7.2

TITLE:

Send DAN

INPUT/OUTPUT:

LDT\_DAN\_Status : data\_out

DAA\_Received : data\_out

DAN : data\_out

DAA\_Status : data\_out

DAN : data\_in

DAA : data\_in

Transfer\_State : data\_in

Valid\_LDT\_Parms : data\_in

BODY:

Description of Process

This function sends a data availability notification (DAN) to LP DAAC and receives a subsequent DAN acknowledgment (DAA) from LP DAAC in response to each DAN received. Upon receipt of DAA, this function verifies the received DAA to identify the status of data transfer scheduling and DAN errors and stores the DAA status information into LDT\_DAN\_Info and File\_Transfer\_Info datastores.

The function can be enabled or disabled by the Transfer\_State flag which is stored in LDT\_DAN\_Transfer\_State datastore. The DAN is written to (send) or read from (resend) in the DAN\_List. This function notifies the MACS of either the successful and unsuccessful transmission of the DAN.

Assumptions

Preconditions

None

Postconditions

LDT\_DAN\_Info and LDT\_File\_Trans\_Info datastores will be updated accordingly.

Constraints

Message header will be defined in the ICD between LPS and LP DAAC.

Functional Breakdown

Upon receipt of a constructed DAN from either Generate\_DAN function or Resend\_DANs function, this function shall store the DAN into the DAN\_List datastore.

This function shall check the Transfer\_State flag to see if the DAN transfer operation is disabled or not. If the DAN transfer function is disabled, this function shall suspend the DAN for further processing; otherwise, this function shall send the DAN over to LP DAAC.

After a DAN is sent over to LP DAAC, this function shall wait for a subsequent Data Availability Acknowledgement (DAA) from LP DAAC.

Once a corresponding DAA is received, this function shall verify the received DAA to identify the DAA\_Status of data transfer scheduling and DAN errors.

This function shall store the DAA\_Status into the LDT\_DAN\_Info datastore for the corresponding DAN. It shall also store the DAA\_Status into the LDT\_File\_Trans\_Info datastore for the file transfer summary report. Finally, this function shall report the DAN processing status to MACS.

**Reusability**

The communication protocol software was reused from Pacor II/DDF.

NAME:

7.3

TITLE:

Receive DDN

INPUT/OUTPUT:

LDT\_DDND\_Status : data\_out

Health\_Message : data\_out

DDA : data\_out

DDN\_Status : data\_out

Contact\_Sequence\_Id : data\_out

Transfer\_Status : data\_out

DDN : data\_in

Valid\_LDT\_Parms : data\_in

LDT\_Stop\_DDND\_Drct : data\_in

Sub\_Intv\_List : data\_in

LPS\_Configuration : data\_in

Current\_Time : data\_in

BODY:

Description of Process

This function creates a server to receive the Data Delivery Notice for a corresponding DAN. Once a DDN is received, it verifies the DDN to identify the file transfer problems occurred in LP DAAC for a contact period LOR file transfer. This function also generates a Data Delivery Acknowledgement (DDA) to LP DAAC to acknowledge receipt of the DDN. It updates the File\_Trans\_Info datastore accordingly. It notifies MACS that a DDN was received and marks each file as having been transferred.

Assumptions

Preconditions

None

Postconditions

LDT\_File\_Trans\_Info datastore will be updated according to the DDN.

Constraints

DDN format will be defined in the ICD between LPS and LP DAAC.

Functional Breakdown

This function retrieves the LDTS setup parameters from Valid\_LDT\_Parms datastore.

This function creates a server based on the retrieved LDTS parameters to receive DDN message from LP DAAC

Upon receipt of DDN message from LP DAAC, this function will verify the received DDN message to identify the DDN\_Status for data transfer and archival.

This function retrieves the subintervals for current contact period to be used to determine the file group transfer status

This function shall update the file transfer status as a subinterval file group and a contact period basis in the LDT\_File\_Trans\_Info, and as an individual LOR file basis in LPS\_File\_Info datastores according to DDN\_Status

This function shall generate a Data Delivery Acknowledgement (DDA) to acknowledge the receipt of DDN message and send it to LP DAAC.

This function shall report a health message periodically to LPS operator.

This function shall send a DDN received message to MACS whenever a DDN message is received from LP DAAC

Upon receipt of Stop\_DDN request, this function shall finish the current processed DDN and terminate itself.

If a no-error DDN message is received, this function shall invoke the Delete\_LPS\_Files function to delete current contact period LOR files automatically.

#### Reusability

Communication protocol software was reused from Pacor II/DDF project.

NAME:

7.4

TITLE:

Resend DANs

INPUT/OUTPUT:

DAN : data\_out

DAN : data\_in

DAN\_Status : data\_in

LDT\_Resend\_DAN\_Drct : data\_in

BODY:

Description of Process

This function sends all suspended DANs or resend a failed DAN to LP DAAC after the file transfer operation is resumed or the DAN error occurred in the received DAA from LP DAAC.

Assumptions

Preconditions

None

Postconditions

None

Constraints

Contact\_Id will be the primary identifier for the suspended DANs the failed DANs.

COTS File transfer protocol (as per the ICD between LPS and LP DAAC) will be used.

Functional Breakdown

Upon receipt of LDT\_Resend\_DAN\_Drct request, this function shall search all DANs from the DAN\_List datastore that were suspended prior to this function is invoked.

After all suspended DANs are identified, this function shall invoke the Send\_DAN function to send the DAN over to LP DAAC one by one.

If a LDT\_Resend\_DAN\_Drct request is used to resend a failed DAN, this function shall identify the failed DAN in the DAN\_List datastore and reconstruct a new DAN with DAN errors corrected. It then invokes the Send\_DAN function to send the new DAN to LP DAAC.

Reusability

None.

NAME:

7.5

TITLE:

Delete LPS Files

INPUT/OUTPUT:

Deletion\_Status : data\_out

Browse\_File : data\_out

PCD\_File : data\_out

Metadata\_File : data\_out

MSCD\_File : data\_out

Cal\_File : data\_out

Band\_File : data\_out

LDT\_Delete\_Files\_Status : data\_out

LDT\_Delete\_Files\_Drct : data\_in

Transfer\_Status : data\_in

Contact\_Sequence\_Id : data\_in

Retain\_Status : data\_in

BODY:

Description of Process

This function deletes LPS LOR files for a contact period that have been successfully transferred to the LP DAAC unless the files have been marked for retention at the request of the MACS (Retain\_LPS\_Files). The LDT\_File\_Trans\_Info and LPS\_File\_Info datastores are updated to record the file deletions.

Assumptions

Preconditions

None

Postconditions

LPS files identified by MACS have been deleted.

LDT\_File\_Trans\_Info, LPS\_File\_Info are updated to record file deletions.

Constraints

None.

Functional Breakdown

Upon receipt of the LDT\_Delete\_Files\_Drct request from MACS or invoked by the Receive\_DDN process, this function shall check the File\_Transfer status from the LDT\_File\_Trans\_Info datastore.

If current contact period LOR files have been successfully transferred to LP DAAC, this function shall delete the contact period LOR files if current contact period LOR files are not marked for retention.

This function shall update the LDT\_File\_Trans\_Info and LPS\_File\_Info datastore to reflect the file Deletion\_Status.

This function shall generate a File\_Deleted message to MACS after a contact period files have been deleted.

Reusability

None.

NAME:

7.6

TITLE:

Retain LPS Files

INPUT/OUTPUT:

LDT\_Retain\_Files\_Status : data\_out

Retain\_Status : data\_out

LDT\_Retain\_Files\_Drct : data\_in

BODY:

Description of Process

This function, in response to the LDT\_Retain\_Files\_Drct request from the MACS, marks all contact period LOR files to be retained.

Assumptions

Preconditions

None

Postconditions

None

Constraints

None

Functional Breakdown

This function shall extract the Contact\_Id from LDT\_Retain\_Files\_Drct request message.

This function shall mark all files in LDT\_File\_Trans\_Info associated with current Contact\_Id as being marked for retention (via File\_Status).

If there are no such files in LDT\_File\_Trans\_Info, this function shall send a Files\_Not\_Found message to MACS.

Reusability

None.

NAME:

7.7

TITLE:

Control Send DAN

INPUT/OUTPUT:

LDT\_DAN\_Transfer\_State : data\_inout

LDT\_Enable\_File\_Xfer\_Drct : data\_in

LDT\_Disable\_File\_Xfer\_Drct : data\_in

BODY:

Description of Process

This function maintains a flag (Transfer\_Files) that is used to enable or disable sending DAN(s) to LP DAAC.

Assumptions

Preconditions

None.

Postconditions

None.

Constraints

None.

Functional Breakdown

If there is a problem to prevent LPS from sending DAN over to LP DAAC and a LDT\_Disable\_File\_Xfer\_Drct is received from MACS, this function shall disable the file transfer operation.

Otherwise, this function shall enable the file transfer operation between LPS and LP DAAC after it receives a LDT\_Enable\_File\_Xfer\_Drct from MACS.

This function shall store the Transfer\_State in LDT\_DAN\_Transfer\_State datastore.

Reusability

None.

---

#### 4.10.2 Performance Requirements

The following list summarizes the performance requirements allocated to the LDTS:

- 4.10.2.1 The LDTS software on each LPS string shall provide the capability to transfer the equivalent of any combination of the format 1 and format 2 portions of 125 Landsat 7 ETM+ scenes of wideband data per day (approximately 25 to 30 gigabytes per day).
- 4.10.2.2 The LDTS software on each LPS string shall provide the capability to execute concurrently with all other LPS subsystems.
- 4.10.2.2.1 The LDTS software on each LPS string shall output a DAN within 240 seconds of the time of receiving all required input.
- 4.10.2.3 The LDTS software on each LPS string shall provide the capability to reprocess a maximum of 10 percent of a string's daily input volume of wideband data (approximately 12.5 scenes or 2.5 to 3 gigabytes per day).
- 4.10.2.4 The LDTS software on each LPS string shall maintain data processing throughput performance for all Landsat 7 raw wideband data received with a BER of one bit error in  $10^5$  bits, without loss of level-zero processed data and without retransmission.
- 4.10.2.5 The LDTS software on each LPS string shall provide the capability to transfer the string's daily volume of LPS output files to the LP DAAC at an average aggregate rate of 10 Mbps.
- 4.10.2.6 The LDTS software on each LPS string shall output any periodic processing status information that it generates with a maximum latency of 30 seconds between output.

---

## Section 5—Database Analysis

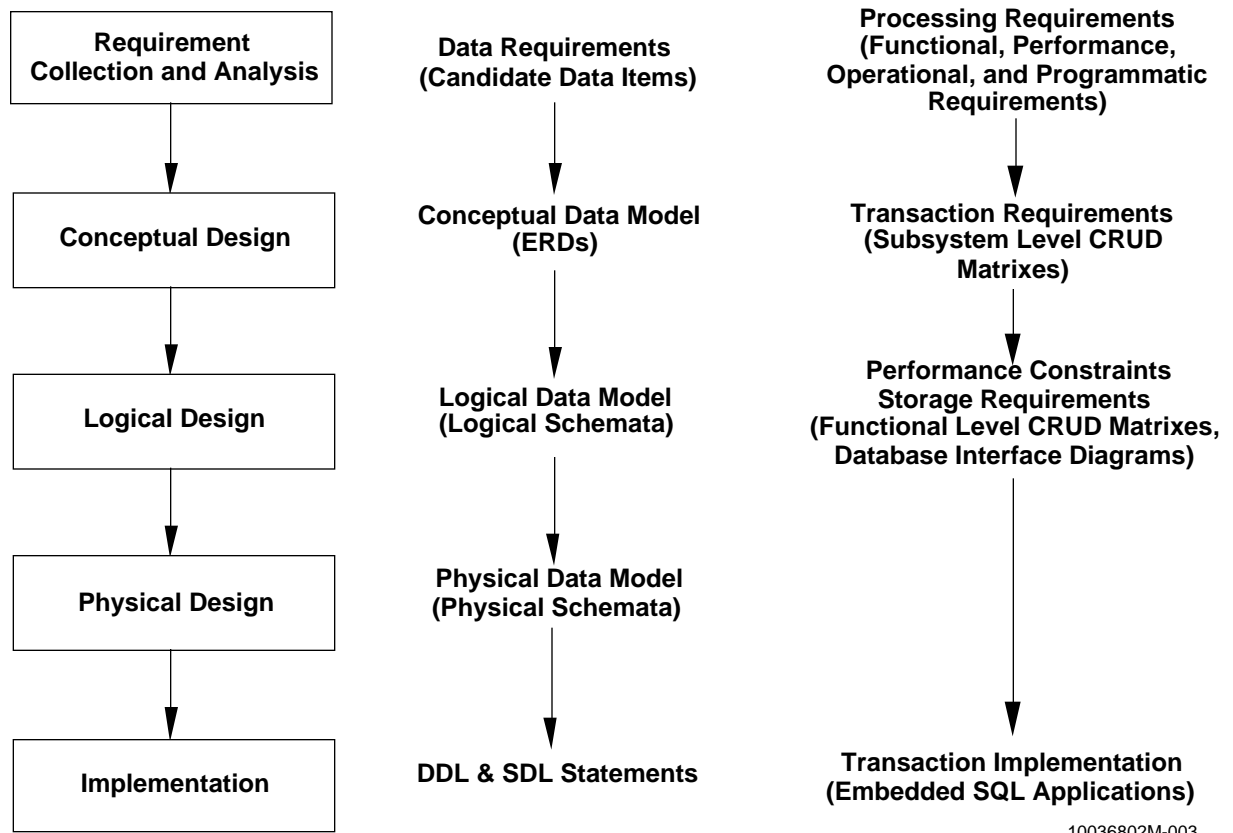
This section presents the results of LPS database design during the software requirement definition phase. This phase includes requirement collection and analysis, conceptual design, and preliminary logical design. The LPS database design follows the standard database methodology that encompasses five major steps (Figure 5–1). The methodology is consistent with *SEAS System Development Methodology*. The design process consists of two parallel activities at each step:

- Data view – the design of data content and structure of the database
- Process view – the design of processing and software application

Both views are described for each step of the process.

1. **Requirement Collection and Analysis** – This step identifies and analyzes the intended uses of the database. From the data point of view, this step identifies the LPS information to be stored in the database. From the processing perspective, this step analyzes the functional, performance, operational, and programmatic requirements of the LPS database. This step is completed during the software requirement definition phase.
2. **Conceptual Design** – This step creates a high-level data model that is DBMS independent. From the data point of view, this step examines the data requirements resulting from step 1 and produces a conceptual database model that consists of entity relationship diagrams (ERDs). From the processing perspective, this step examines the interaction between LPS subsystems and the database and produces subsystem-level Create, Retrieve, Update, and Delete (CRUD) matrixes. This step is completed during the software requirement definition phase.
3. **Logical Design** – This step creates a logical data model for a relational DBMS. From the data point of view, this step creates logical schemata and constraints to represent entities and relationships from the ERDs. Principles of normalization are applied to create well-structured schemata. From the processing perspective, this step produces functional CRUD matrixes and database interface diagrams that describe the interaction between main processes within a subsystem and the database. This step also performs a preliminary analysis of data usage, performance constraints, and storage requirements. Part of this step is

completed during the software requirement definition phase, and the remainder is continued in the next phase.



10036802M-003

**Figure 5-1. LPS Database Design Approach**

4. **Physical Design** – This step selects specific storage structures and access methods for the database. From the data point of view, this step determines storage structures such as tables, views, indexes, and organizations of database information. From the processing perspective, this step further analyzes data usage and performance constraints, which leads to the determination of access methods.
5. **Implementation** – This step implements the database. From the data point of view, this step physically creates the LPS databases, tables, views, and indexes through Data Definition Language (DDL) and Storage Definition Language (SDL) statements. From the processing perspective, this step implements database applications. Database transactions are examined and corresponding program code with embedded SQL commands are

written and tested. Once the transactions are ready, the database will be populated.

The Cadre/Teamwork Information Modeling (IM) tool is used during the design process. Cadre/Teamwork IM is an integrated toolset that helps database engineers model the entities, relationships, and attributes of all LPS data at the conceptual level. The tool is used to

- Create an LPS conceptual model consisting of entity relationship diagrams.
- Ensure the completeness and consistency of database design.
- Generate code to create database tables and enforce integrity constraints.
- Support documentation production.
- Simplify the maintenance effort by easing impact analysis and change implementation.
- Enhance configuration management by maintaining baselines.

---

## 5.1 Requirement Analysis and Conceptual Design

The purpose of database requirements analysis is to identify and analyze the intended uses of the database. This includes identifying the information to be stored in the database and analyzing the functional, performance, operational, and programmatic requirements of the database. Conceptual design creates a high-level data model that consists of ERDs and examines the interaction between LPS subsystems and the database. The following subsections present the results of the requirement analysis and conceptual design phase.

---

### 5.1.1 Functional Requirement Analysis

This section presents a functional overview of the information tracking requirements for the LPS. The functions are described in terms of their interactions with the database. The functions are based on the LPS subsystems and functions presented in Section 3 of *Landsat 7 Processing System (LPS) System Design Specification (SDS)*.

#### **5.1.1.1 Raw Data Capture Subsystem**

The RDCS captures and manages raw wideband data received from the LGS. Each LGS channel's data stream for a contact period is captured to the online disk. The data set is subsequently copied to removable media. On request, the RDCS retrieves a raw wideband data set from removable media and returns it to the online storage for reprocessing. Also on request, the RDCS generates a data receive summary for a specified raw wideband data set. The interactions between the RDCS and the LPS database are as follows:

- The RDCS retrieves the LPS configuration parameters (such as the LPS hardware string identification and the LGS channel associated with the LPS string) from the database when the RDCS starts up.
- The RDCS captures a raw wideband data bitstream for a single channel from the LGS and outputs the stream as a byte stream data set to an online raw wideband datastore. At the end of the contact period, the subsystem collects and stores a contact summary describing the contact in the database.
- On request, the RDCS queries the database and generates a contact data receive summary for displaying and printing.
- On request, the RDCS retrieves the requested raw wideband data from removable media to the online storage for reprocessing. If not already present, contact summary information is collected and stored in the database.

#### **5.1.1.2 Raw Data Processing Subsystem**

The RDPS inputs the rate-buffered wideband data from the online datastore and performs CCSDS AOS Grade 3 service on the received CADUs. This subsystem performs frame synchronization and PRN decoding of the received CADUs, CRC on VCDUs, and RS error detection and correction of VCDU headers. It also performs the BCH error detection and correction processing. The RDPS collects and generates raw data processing accounting and subsystem status information. The interactions between the RDPS and the LPS database are as follows:

- The RDPS receives CCSDS processing parameters and thresholds, transforms them into the subsystem's internal format (if necessary), and stores them in the database.

- The RDPS retrieves CCSDS parameters and thresholds for frame synchronization processing. In addition, thresholds also are retrieved and used during the CCSDS AOS Grade 3 service and BCH processing.
- The RDPS processes the raw wideband data set for a contact period. It uses the CCSDS parameters and thresholds retrieved from the database for its processing and error reporting. The RDPS accumulates and collects quality and accounting information on a contact basis and then stores the information in the database.
- On request, the RDPS queries the database and generates return-link quality and accounting summary for displaying and printing.

#### **5.1.1.3 Major Frame Processing Subsystem**

The MFPS processes annotated CADUs and provides the functionality to synchronize the major frames, extract major frame times, deinterleave band data, reverse band data if necessary, and align band data. It also is responsible for generating the calibration and MSCD files. In addition, it determines subinterval boundaries and extracts and provides PCD information to the PCDS. The MFPS also collects and generates LOR accounting and subsystem status information. The interactions between the MFPS and the LPS database are as follows:

- The MFPS retrieves and uses the subinterval threshold to determine subintervals. Data about generated subintervals is stored in the database for use by other subsystems.
- The MFPS retrieves and uses thresholds for error reporting during its processing.
- The MFPS retrieves and uses sensor alignment information for band alignment processing.
- The MFPS collects LOR quality and accounting information on a subinterval and major frame basis. Information collected is stored in the database used by the MACS for metadata generation.

#### **5.1.1.4 PCD Processing Subsystem**

The PCDS processes PCD and is responsible for performing PCD byte majority voting, PCD major frame building, and PCD file generation. In

addition, it identifies scenes using the WRS, calculates Sun azimuth and elevation information, and extracts ETM+ calibration door events. The PCDS also collects and generates PCD accounting and subsystem status information. The interactions between the PCDS and the LPS database are as follows:

- The PCD retrieves and uses the frame quality parameters during the process of assembling PCD cycles.
- The PCD retrieves and uses error report thresholds for error reporting during its processing.
- The PCD retrieves and uses scene calculation parameters and the WRS information to identify WRS scenes and calculate Sun azimuth and elevation. Calculated scene and Sun information is stored in the database and used by other subsystems.
- The PCDS collects quality and accounting information during its processing of PCD data and then stores the information in the database for use by the MACS for metadata generation.

#### **5.1.1.5 Image Data Processing Subsystem**

The IDPS performs general LPS image data processing and generates L0R instrument data files and browse files. It also is responsible for determining cloud coverage on a scene quadrant and full-scene basis and generating an MWD. The IDPS also collects and generates IDPS accounting and subsystem status information. The interactions between the IDPS and the LPS database are as follows:

- The IDPS receives, validates, and stores band parameters for browse and ACCA in the database.
- The IDPS retrieves and uses subinterval information, detector gains and biases, radiometric correction parameters, and band parameters to generate the browse files.
- The IDPS retrieves and uses subinterval information to generate the band files.
- The IDPS retrieves and uses subinterval information, detector gains and biases, radiometric correction parameters, and band parameters for its ACCA processing. ACCA scores are generated and stored in the database to be included in the metadata file.
- The IDPS retrieves and uses MWD parameters.

- The IDPS collects accounting information during its processing and then stores the information in the database for use by the MACS for metadata generation.

#### **5.1.1.6 Management and Control Subsystem**

The MACS provides an interface through which operations personnel can control the system operations, monitor overall system performance, access system accounting information, and manage the configuration parameters and thresholds that are used to drive system processing. The MACS also provides capabilities to generate metadata files and accounting information. The interactions between the MACS and the LPS database are as follows:

- The MACS manages the LPS system configuration parameters in the database. It allows the operator to update the LPS system configuration parameters.
- The MACS allows the operator to update, ingest, and propagate the contact schedules.
- The MACS allows the operator to update and propagate LOR parameters.
- The MACS allows the operator to update LOR thresholds.
- The MACS is responsible for generating metadata files. During the metadata file generation process, the MACS retrieves system configuration parameters, LOR quality and accounting information, PCD quality and accounting information, browse and band file accounting information, ACCA scores, WRS scene information, and subinterval data to generate metadata files on a subinterval basis. Metadata accounting and LPS file information also is generated and stored in the database for subsequent processing.

#### **5.1.1.7 LPS Data Transfer Subsystem**

The LDTS coordinates the transfer of LPS files to the LP DAAC over supported network connections. The LDTS also is responsible for overseeing the LPS file transmission, maintaining the status of all LPS files, managing the LPS output datastore, and accumulating the file transfer summary. The interactions between the LDTS and the LPS database are as follows:

- The LDTS retrieves and uses configuration parameters and LPS file information from the database for DAN generation and DDN processing. The LPS file information is organized on a contact basis.
- The LDTS maintains the status of all LPS files in the database. The status of LPS files is updated frequently to reflect data transfer status from the LP DAAC.
- The LDTS is responsible for managing LPS output data storage. It retrieves the status of LPS files from the database when deciding whether to retain or delete certain LPS files.

---

### 5.1.2 Performance Requirement Analysis

Because all LPS subsystems interact with the database, the performance requirements for LPS subsystems include database performance considerations. Main factors affecting the database performance are discussed in the following subsections.

#### 5.1.2.1 Response Time

The performance of the database is part of the LPS performance. To provide the capability to process received wideband data at a minimum rate of 7.5 Mbps, the database access time must be minimized and optimized. To achieve this, the LPS database must support the following capabilities:

- Storage Optimization – Distribution of database tables over different disk drives where speed of disk reads and writes is crucial.
- Indexing – Access structures that are used by applications to speed up the retrieval of records in response to certain search conditions.
- Viewing – Virtual tables in which data from underlying base tables are combined so that applications can work with just one virtual table instead of the several or more complete base tables.
- Query optimization – Heuristic and cost-based query optimization mechanisms to improve the efficiency of query execution.
- Stored Procedures – Named and precompiled set of SQL statements that are stored in the server's data dictionary and can be executed by applications through names.

- Denormalization – A process in which columns belonging to one table are redundantly defined in another table to reduce or eliminate the need to query the original table.

#### **5.1.2.2 Reliability, Maintainability, and Availability**

The RMA of the database is part of the LPS RMA. To support LPS operations 24 hours per day, 7 days per week, to provide an operational availability ( $A_0$ ) of 0.96 or better; achieve a mean time to restore capability of 4 hours or better; and comply with the requirement of not exceeding twice the required mean time to restore in 99 percent of failure occurrences, the LPS DBMS must support the following capabilities:

- Online backups and recovery
- Online archiving
- Mirroring of critical database files (TBD)

#### **5.1.2.3 Data Integrity**

Data integrity is a data quality issue. Four categories of data integrity must be specified and enforced through appropriate mechanisms:

1. Entity integrity – Integrity that guarantees all primary key attribute values are not null. This capability should be supported by the DBMS.
2. Domain integrity – Integrity that enforces attribute values to adhere to the underlying application domain definitions. This capability should be supported by the DBMS.
3. Referential integrity – Integrity that guarantees the existence and correctness of the required relationship between two entities. This capability should be supported by the DBMS.
4. LPS application integrity – Integrity that protects the validity of attribute values. This capability is not provided by the DBMS and requires application software.

---

### 5.1.3 Programmatic Requirement Analysis

Each LPS subsystem interacts with the database. To permit efficient and effective programming, the LPS DBMS will

- Be a relational database.
- Comply with ANSI/ISO SQL.
- Interface with the C language through embedded SQL.
- Provide capabilities to generate and access stored procedures.

Automated routines and user interface facilities will be implemented to support the population of LPS database.

---

### 5.1.4 High-Level Entity Relationship Model

The LPS entity relationship (ER) model is a conceptual representation of the data for LPS application. The ER model is expressed in terms of entities in the LPS environment, the relationships or associations among those entities, and the properties of the entities and their relationships. Entities represent data items that play a functional role in the LPS application and have their own set of attributes. Table 5–1 describes each LPS entity identified during the conceptual design process. The attributes of each entity are analyzed and described in Section 5.2. Note that new entities may be added and existing ones modified, merged, or split as the project progresses and the definitions of entities become more precise.

The LPS ER model is expressed as ERDs, which are graphical representations of the ER model as illustrated in Figure 5–2. In addition to entities described previously, relationships also are included in the diagram. The relationships represent the association between the instances of one or more entities that are of interest to the LPS. A cardinality is associated with each relationship. The cardinality describes the number of instances of one entity that can be associated with each instance of the entity to which it relates. For instance, there is more than one scene information (PCD\_Scene\_Acct) for each subinterval (Sub\_Intv), while there is only one MFPS accounting information (MFP\_Acct) for each subinterval.

**Table 5-1. LPS Entity Descriptions (1 of 2)**

Entity Name	Entity Description
Contact_Schedules	A set of contact periods containing the start and stop times when the Landsat 7 spacecraft downlinks the wideband data to the LGS. The schedule is coming from the LGS in a hardcopy form.
IDP_Acct	Aggregate accounting information for the IDPS that includes ETM+ band and ACCA accounting information on a scene basis.
LDT_Output_File_Info	All state information about LPS output files that are of concern to LDTS.
LPS_Configuration	The set of parameters used to configure the LPS. Some parameters are used when the system starts up while others are used during the processing.
MFP_Acct	An aggregate accounting information from the MFPS that includes the LOR quality and accounting information on a subinterval basis.
PCD_Acct	An aggregate accounting information from the PCDS that includes the processed PCD quality and accounting information on a subinterval basis.
RDC_Acct	Raw data capture accounting information on a contact basis.
RDP_Acct	An aggregate accounting information from the RDPS which includes return link quality and accounting information on a contact basis.
Sub_Intv	A list of subinterval information generated by the MFPS and used by the PCDS, IDPS, and MACS to generate LPS files.
Valid_Band_Parms	Aggregate information that includes parameters and reduction data for browse processing.
Valid_CCSDS_Parms	A list of parameters that controls CCSDS frame synchronization and bit slip correction.
Valid_MFP_Parms	Validated MFPS setup parameters.
Valid_MFP_Thres	Validated MFPS threshold values.
Valid_MWD_Parms	Set of parameters that configure the MWD.
Valid_PCD_Parms	Validated PCD parameters used in processing PCD data.
Valid_PCD_Thres	Validated PCD threshold parameters used in processing PCD data.
Valid_RDP_Thres	Validated RDP processing thresholds.

**Table 5-1. LPS Entity Descriptions (2 of 2)**

Entity Name	Entity Description
Valid_Scene_Parms	Validated general mission information and parameters provided by the IAS and used to calculate the longitude, latitude, the WRS scene identifier and Sun elevation and azimuth.
Valid_WRS_Parms	Validated WRS table containing the information for each WRS scene.



---

## Section 6—User Interface

The LPS user interface consists of all elements of interaction between the LPS and the operator. For the LPS, this interaction will probably be some combination of system-level commands and Oracle user interface products. There is the possibility that existing products, such as a process manager (DPCP as described in Section 6.3.2), will be used.

Concurrent development of the software requirements and the user interface offers specific advantages. By examining the interactions expected between the operators and the system, potential problems can be avoided and an understanding of what the subsystems must do to perform their function can be aided. Another important benefit to beginning work on the user interface is that software drivers may be uncovered by closely inspecting how the system will be required to operate. Finally, early effort can lead to a preliminary user interface, offering the chance for users to provide input into the design decisions of both the final user interface and the application software.

Preliminary analysis of user interface has covered the areas of task analysis, performance goal setting, and user interface mockup. The following subsections present the results of these studies

---

### 6.1 Task Analysis

Task analysis provides a conceptual framework from which to approach the design of the user interface. It explains drivers and limitations on what the user interface will or will not be.

---

#### 6.1.1 Drivers

The operator is required to set up, test, monitor, and control the LPS.

Each LPS string is physically and logically independent. Each string must have its own user interface.

Operations can be performed on several contacts at one time. The processing of contact 1 can still be going on while contact 2 needs to be captured.

---

### 6.1.2 Constraints

The budget for the user interface is very limited. Many choices will be based on budgetary constraints.

---

### 6.1.3 Assumptions

No network interface will be available for input of schedules, parameters, etc. between LPS and external systems.

No security will be provided other than what is available from the UNIX shell and from Oracle.

There will be only one type of user for the LPS. This user type is classified as an operator.

LPS operators will be capable of using the operating system to perform some user interface functions. No elaborate shell program is needed to buffer the operator from UNIX.

No long-term or trend reporting is required.

---

### 6.1.4 Decisions

The user interface will be developed as some combination of UNIX shell commands, Oracle Forms and Oracle Reports.

There is no automated coordination between strings. This implies that there is no systemwide reporting.

---

### 6.1.5 User Interface Event List

This section contains a complete list of all known interactions between the LPS and the operator.

#### SYSTEM CONFIGURATION

F&PS	Event	Subsystem	Data Dictionary
3.1.14	Configure LPS (normal/fallback)	System level	None

**THRESHOLDS**

<b>F&amp;PS</b>	<b>Event</b>	<b>Subsystem</b>	<b>Data Dictionary</b>
3.3.6.6	Input RDPS thresholds	RDPS	RDP_Thresholds
3.3.6.6	Input MFPS thresholds	MFPS	MFP_Thresholds
3.3.6.6	Input PCDS thresholds	PCDS	PCD_Thresholds

**PARAMETERS**

<b>F&amp;PS</b>	<b>Event</b>	<b>Subsystem</b>	<b>Data Dictionary</b>
3.3.6.1	Input LPS configuration	MACS	LPS_Configuration
3.2.1	Input contact schedules (from MOC)	MACS	Contact_Schedules
derived	Input MFP parameters	MFPS	MFP_Parms
3.3.2.22	Input sensor align tables (from IAS)	MFPS	Sensor_Alignment_Info
3.3.2.1	Input CCSDS AOS grade 3 parameters	RDPS	RDP_CCSDS_Parms
3.3.3.2	Input browse monochrome band	IDPS	IDP_Band_Parms
3.3.3.3	Input browse multiband 1, 2, and 3	IDPS	IDP_Band_Parms
3.3.4.10	Input ACCA comparison values	IDPS	IDP_Band_Parms
derived	Input PCD parameters	PCDS	PCD_Parms

**TEST**

<b>F&amp;PS</b>	<b>Event</b>	<b>Subsystem</b>	<b>Data Dictionary</b>
3.1.10.6	Test functions and external interfaces	System level	None
3.1.10.7	Execute diagnostic tests	System level	None
3.1.10.8	Support end-to-end testing of LPS functions	System level	None
3.1.19	Read test points to verify proper operation	System level	None

**CONTROL**

<b>F&amp;PS</b>	<b>Event</b>	<b>Subsystem</b>	<b>Data Dictionary</b>
3.1.10.1	Start up LPS	IRIX	None
3.1.10.2	Shut down LPS	IRIX	None
3.1.11	Control LPS operations		
3.3.6.9a	Start capture of wideband data	RDACS	RDC_Capture_Drct
3.3.6.9a	Stop capture of wideband data	RDACS	RDC_Capture_Drct
3.3.1.7	Start copy from disk to tape	RDACS	RDC_Save_Drct
derived	Stop copy from disk to tape	RDACS	RDC_Save_Drct
3.3.1.9	Start copy from tape to disk	RDACS	RDC_Restage_Drct
derived	Stop copy from tape to disk	RDACS	RDC_Restage_Drct
3.3.6.9b	Start LOR processing	RDPS	RDP_Process_Drct
3.3.6.9b	Stop LOR processing	IRIX kill	None
3.3.6.8	Manually override automated functions	IRIX kill	None

**MONITORING**

<b>F&amp;PS</b>	<b>Event</b>	<b>Subsystem</b>	<b>Data Dictionary</b>
3.1.12	Monitor LPS operations	All (MACS)	Assorted error messages
3.3.6.7	Report error threshold exceeded	All	Assorted error messages
3.3.6.7	Report result threshold exceeded	All	Assorted error messages
3.1.10.3	Report error messages	All (MACS)	Assorted error messages
3.1.10.4	Isolate system faults	MACS, IRIX	LPS_Journal
3.1.10.5	Recover from system faults	MACS, IRIX	LPS_Journal
derived	Examine LPS_Journal	Text Editor	LPS_Status

**FILE MANAGEMENT**

<b>F&amp;PS</b>	<b>Event</b>	<b>Subsystem</b>	<b>Data Dictionary</b>
3.3.6.9c	Enable file transfer	MACS, LDTS	LDT_Enable_File_Xfer_Drct
3.3.6.9c	Disable file transfer	MACS, LDTS	LDT_Disable_File_Xfer_Drct
derived	Delete raw data input file	RDSCS	RDC_Delete_Drct
3.3.5.5	Delete output files on contact basis	LDTS	LDT_Delete_Files_Drct
3.3.5.6	Retain output files on contact basis	LDTS	LDT_Retain_Files_Drct
3.3.2.8	Examine CADU CCSDS trouble files	IRIX	N/A
3.3.2.10	Examine CADU BCH trouble files	Text Editor	N/A
derived	Resend DAN to the LP DAAC	LDTS, MACS	LDT_Resend_DAN_Drct

**REPORTS**

<b>F&amp;PS</b>	<b>Event</b>	<b>Subsystem</b>	<b>Data Dictionary</b>
3.3.1.10.1	Display wideband data receive summary	RDSCS	RDC_Rpt_Data_Capture_Sum_Drct
3.3.1.10.1	Print wideband data receive summary	RDSCS	RDC_Rpt_Data_Capture_Sum_Drct
3.3.6.4	Display return-link quality and accounting data (contact)	RDPS	RDP_Rpt_Return_Link_QA_Drct
3.3.6.4.1	Print return-link quality and accounting data (contact)	RDPS	RDP_Rpt_Return_Link_QA_Drct
3.3.6.4	Display LOR quality and accounting data (subinterval)	MFPS	MFP_Rpt_LOR_QA_Drct
3.3.6.4.1	Print LOR quality and accounting data (subinterval)	MFPS	MFP_Rpt_LOR_QA_Drct
3.3.6.5	Display transfer summary (contact)	LDTS	LDT_Rpt_File_Xfer_Sum_Drct
3.3.6.5.1	Print transfer summary (contact)	LDTS	LDT_Rpt_File_Xfer_Sum_Drct

---

## **6.2 User Interface Goals**

This section is commonly used to define specific quantitative goals defining minimal acceptable user interface performance. Due to the limited budget of the LPS, it is undesirable to place hard restrictions on items such as response time, when a slightly relaxed requirement could produce a user interface at a much lower cost. The LPS will seek to develop a user interface which is responsive, easy to use, and maximizes efficient operations.

---

## **6.3 User Interface Mockup**

The user interface consists of three different types of commands: operating system, reusable COTS software, and Oracle Forms and Oracle Reports.

---

### **6.3.1 Operating System**

UNIX commands will be used to start the LPS on each string.

The UNIX tar command can satisfy the need to start and stop the copy to short-term storage.

The operating system will be used to set the priority of LPS processes.

---

### **6.3.2 Reusability**

If applicable and practical, preexisting software will be used for the LPS user interface. These reuse sources are described in Section 3.

One possible source of such software is NASA's Ground Operations Technology Testbed (Code 520).

The first possible reuse item identified is the DPCP, which enables an operator to start and monitor a set of processes running on one or more host computers. The second possible reuse tool is the DAMT, which may help to analyze the performance of the system.

Another possible source of reuse is the SDPF CIS.

## Appendix—Requirements Traceability

This appendix presents the LPS requirements traceability. Table A–1 shows the mapping between the lowest level processes in the DFDs and the F&PS requirements. Table A–2 shows the mapping between F&PS requirements and the lowest level processes in the DFDs.

**Table A-1. Software to System Requirements Traceability**

F&PS Requirement	Software Requirement No.	Software Requirement Name	LPS Software Module
3.1.4, 3.1.6, 3.1.11, 3.2.1, 3.3.1.1, 3.3.1.2, 3.3.1.3, 3.3.1.4, 3.3.1.12, 3.3.2.13, 3.3.4.12	1.1	Receive Raw Wideband Data	rdc_Capture rdc_UpdRDCAcct
3.1.4, 3.3.1.6, 3.3.1.7	1.2	Save Raw Wideband Data	rdc_Save
3.3.1.9	1.3	Restage Raw Wideband Data	rdc_RestageCptr
3.3.1.1, 3.3.1.2, 3.3.1.3, 3.3.1.4	1.4	Delete Raw Wideband Data	rdc_DeleteRDCFiles
3.3.1.5, 3.3.1.6	1.5	Replay Raw Wideband Data	rdc_Transmit
3.3.1.10	1.6	Generate Media Label	rdc_GenLabel
	2.1	Synchronize CCSDS Frame	
3.1.4, 3.1.6, 3.1.7, 3.3.2.2, 3.3.2.3, 3.3.2.4, 3.3.2.13, 3.3.2.26, 3.3.2.27, 3.3.4.13, 3.3.6.7	2.1.1	Perform SCLF Synchronization	rdp_MainExtractCADU rdp_MainFSync rdp_MainTranToShared
3.1.4, 3.3.2.6	2.1.2	Align Bytes	rdp_MainExtractCADU
3.1.4, 3.1.6, 3.3.2.3, 3.3.2.5, 3.3.2.13	2.1.3	Deinvert Data	rdp_MainExtractCADU
3.1.4, 3.3.2.7	2.1.4	Perform PN Decode	rdp_MainExtractCADU

	2.2	Process CCSDS Grade 3	
3.1.4, 3.1.6, 3.1.7, 3.3.2.1, 3.3.2.8, 3.3.2.13, 3.3.2.26, 3.3.2.27, 3.3.4.13, 3.3.6.7	2.2.1	Perform CRC Check	rdp_MainValidateCADU rdp_MainGenerateOutput
3.1.4, 3.1.6, 3.1.7, 3.3.2.1, 3.3.2.8, 3.3.2.13, 3.3.2.26, 3.3.2.27, 3.3.4.13, 3.3.6.7	2.2.2	Perform RS_EDAC Check	rdp_MainValidateCADU rdp_MainGenerateOutput
3.1.4, 3.1.6, 3.1.7, 3.3.2.9, 3.3.2.9.1, 3.3.2.10, 3.3.2.13, 3.3.2.26, 3.3.2.27, 3.3.4.13, 3.3.6.7	2.3	Decode BCH	rdp_BCHDecode rdp_MainGenerateOutput
3.1.4, 3.3.2.11	2.4	Annotate VCID Change	rdp_MainValidateCADU
3.3.2.11, 3.3.2.12, 3.3.2.19	3.1	Identify VCDUs	mfp_MainIdentifyMjfSet
3.3.2.28, 3.3.4.16	3.2	Extract PCD	mfp_MainPcdStatusProc
	3.3	Parse Major Frame	
3.3.2.14, 3.3.2.15	3.3.1	Identify Major Frames	mfp_MainValidateMjf
3.1.7, 3.3.2.26	3.3.2	Extract Major Frame Time	mfp_MainValidateMjf
3.1.7, 3.3.2.26, 3.3.2.27, 3.3.6.7	3.3.3	Collect VCDU Quality and Accounting	mfp_MainValidateMjf
3.3.2.11, 3.3.2.23	3.3.4	Determine Subintervals	mfp_MainDetermineSub
	3.4	Generate Band Data	mfp_MainBandGen
3.3.2.14, 3.3.2.16, 3.3.2.17, 3.3.2.18	3.4.1	Deinterleave and Reverse Bands	mfp_MainBandGen
3.3.2.19, 3.3.2.22	3.4.2	Align Bands	mfp_MainBandGen

	3.5	Extract Calibration and MSCD	mfp_L0RFilesGen
3.1.5, 3.3.2.24, 3.3.2.25, 3.3.2.28, 3.3.4.12	3.5.1	Create MSCD File	mfp_L0RFilesGen
3.1.5, 3.3.2.24, 3.3.2.25, 3.3.4.12	3.5.2	Create Calibration File	mfp_L0RFilesGen
3.3.2.14, 3.3.2.20	3.5.3	Extract MSCD Data	mfp_L0RFilesGen
3.3.2.14, 3.3.2.21	3.5.4	Extract Calibration Data	mfp_L0RFilesGen
3.1.6, 3.1.7, 3.3.2.13, 3.3.2.26, 3.3.4.13	3.6	Collect Quality and Accounting	mfp_MainQASubGen
	4.1	Assemble PCD Frames	
3.3.4.1	4.1.1	Locate PCD Sequences	pcd_MainDeterminePcdWord
3.3.4.1	4.1.2	Sync on Unpacked PCD	pcd_MainDeterminePcdWord
3.3.4.1, 3.3.4.15	4.1.3	Pack PCD	pcd_MainDeterminePcdWord
3.3.4.1, 3.3.4.14	4.1.4	Assemble PCD Minor Frames	pcd_MainBuildCycle
3.3.4.3, 3.3.4.14	4.1.5	Assemble PCD Major Frames	pcd_MainBuildCycle
3.3.4.3, 3.3.4.14	4.1.6	Assign PCD Major Frame Times	pcd_MainBuildCycle
3.3.6.7	4.1.7	Issue Missing Words Alarm	pcd_MainBuildCycle
3.3.6.7	4.1.8	Issue Majority Vote Failure Alarm	pcd_MainBuildCycle
3.3.4.5, 3.3.4.12, 3.3.4.14, 3.3.4.15	4.2	Summarize PCD Quality	pcd_MainCreatePcdFile
3.3.4.12	4.3	Extract Bands Present	pcd_MainExtractCycleInfo
3.3.4.7, 3.3.4.15, 3.3.4.16	4.4	Extract Scacecraft Position	pcd_MainDetermineScenes

3.3.4.7, 3.3.4.15, 3.3.4.16	4.5	Validate Space- craft Position	pcd_MainDetermineScenes
	4.6	Identify WRS Scenes	
3.3.4.7, 3.3.4.16	4.6.1	Compute Look Points	pcd_MainDetermineScenes
3.3.4.7, 3.3.4.16, 4.3.5	4.6.2	Locate Scene Centers	pcd_MainDetermineScenes
3.3.4.7, 3.3.4.16, 4.3.5	4.6.3	Compute Corner Positions	pcd_MainDetermineScenes
3.3.4.7, 3.3.4.16, 4.3.5	4.6.4	Compute Horizontal Display Shift	pcd_MainDetermineScenes
3.3.4.7, 3.3.4.16	4.6.5	Calculate Sun Position	pcd_MainDetermineScenes
3.3.4.7, 3.3.4.16	4.6.6	Assemble Scene Messages	pcd_MainDetermineScenes
3.3.2.29, 3.3.4.12, 3.3.4.16	4.6.7	Assemble Scene Metadata	pcd_MainDetermineScenes
3.3.2.29, 3.3.4.16	4.6.8	Report Cal Door Activity	pcd_MainDetermineScenes
3.3.4.4	4.7	Create PCD File	
3.3.4.4	4.7.1	Convert ADS to EU	pcd_MainCreatePcdFile
3.3.4.4	4.7.2	Convert ADS Temperature to EU	pcd_MainCreatePcdFile
3.3.4.4	4.7.3	Convert Gyro to EU	pcd_MainCreatePcdFile
3.3.4.4	4.7.4	Convert Gyro Drift to EU	pcd_MainCreatePcdFile
3.1.5, 3.3.2.24, 3.3.2.25, 3.3.4.2, 3.3.4.4	4.7.5	Create PCD Output Files	pcd_MainCreatePcdFile
3.3.4.4, 3.3.4.12	4.7.6	Create PCD File Names	pcd_MainCreatePcdFile
	5.2	Generate Browse File	

3.3.3.1, 3.3.3.3, 3.3.3.4	5.2.1	Correct and Contrast Stretch Image	idp_Browse
3.1.5, 3.3.2.24, 3.3.3.1, 3.3.3.3, 3.3.3.4, 3.3.3.5, 3.3.4.12, 4.3.4	5.2.2	Reduce Image by Wavelets	idp_Browse
3.1.5, 3.3.2.24, 3.3.2.25, 3.3.4.12, 3.3.4.16	5.3	Generate Band File	idp_Band
	5.4	Perform ACCA	
3.3.4.8, 3.3.4.9	5.4.1	Collect Scene Data	idp_ACCA
3.3.4.8, 3.3.4.9, 3.3.4.10, 3.3.4.16	5.4.2	Generate Cloud Cover Assess- ment	idp_ACCA
3.3.6.10	5.5	Generate Moving Window Display	idp_mwd
3.1.11, 3.3.5.5, 3.3.5.6, 3.3.6.1, 3.3.6.4, 3.3.6.5	6.1	Process LPS Directive	mac_ui_MainMenu
3.1.5, 3.3.2.24, 3.3.4.11, 3.3.4.12, 3.3.4.13, 3.3.4.14, 3.3.4.16	6.2	Generate Metadata	mac_MetaDataGen
3.1.11, 3.3.6.7	6.3	Report LPS Status	mac_ui_MainMonitor
3.2.4, 3.3.4.12, 3.3.6.1	6.4	Ingest Cal File	mac_ui_MainControl
3.1.14, 3.3.6.1, 3.3.6.6, 3.3.6.9	6.5	Modify LOR Parameters and Thresholds	mac_ui_MainSetup
3.2.3, 3.3.6.1	6.6	Ingest Contact Schedule	mac_ui_MainSetup
3.1.10.1, 3.1.10.2, 3.3.6.8	6.7	LPS System Control	mac_ui_MainFileMgt mac_yu_MainControl
3.1.10.4, 3.1.10.5	6.8	Monitor System Faults	mac_ui_MainMonitor

	6.9	Validate Parameters	
3.3.6.1	6.9.1	Validate RDP Parameters	mac_ui_MainSetup
3.3.6.1	6.9.2	Validate MFP Parameters	mac_ui_MainSetup
3.3.6.1	6.9.3	Validate IDP Parameters	mac_ui_MainSetup
3.3.6.1	6.9.4	Validate PCD Parameters	mac_ui_MainSetup
3.3.6.1	6.9.5	Validate RDC LDT Parameters	mac_ui_MainSetup
	6.10	Display or Print LPS Report	
3.3.1.10	6.10.1	Generate Data Receive Summary Report	mac_ui_MainReports
3.1.6, 3.1.7, 3.3.2.13, 3.3.2.26, 3.3.6.2, 3.3.6.3, 3.3.6.4, 3.3.6.4.1	6.10.2	Generate Level 0R QA Report	mac_ui_MainReports
3.3.5.7, 3.3.6.5, 3.3.6.5.1	6.10.3	Generate File Transfer Summary Report	mac_ui_MainReports
3.2.2, 3.3.5.1	7.1	Generate DAN	ldt_CreateDAN
3.2.2, 3.3.5.1	7.2	Send DAN	ldt_SendDAN
3.3.5.3	7.3	Receive DDN	ldt_RcvDDN
3.2.2, 3.3.5.1	7.4	Resend DANs	ldt_RsndSuspDANs
3.3.5.4, 3.3.5.5	7.5	Delete LPS Files	ldt_DeleteFiles
3.3.5.4, 3.3.5.6	7.6	Retain LPS Files	ldt_RetainFiles
3.2.2, 3.3.5.1	7.7	Control Send DAN	ldt_RsndSuspDANs

**Table A-2. System to Software Requirements Traceability**

<b>Software Requirement No.</b>	<b>F&amp;PS Requirement No.</b>
General	3.1.1
General	3.1.2
General	3.1.3
1.1, 1.2, 6.2	3.1.4
3.5.1, 3.5.2, 4.7.5, 5.2.2, 5.3, 6.2	3.1.5
1.1, 2.1.1, 2.1.3, 2.2.1, 2.2.2, 2.3, 3.3.2, 3.3.3, 3.6, 6.10.2	3.1.6
2.1.1, 2.1.3, 2.2.1, 2.2.2, 2.3, 3.3.2, 3.3.3, 3.6, 6.10.2	3.1.7
General	3.1.8
General	3.1.10
6.7	3.1.10.1
6.7	3.1.10.2
General	3.1.10.3
6.8	3.1.10.4
6.8	3.1.10.5
General	3.1.10.6
General	3.1.10.7
General	3.1.10.8
1.1, 6.1, 6.3	3.1.11
6.3	3.1.12
6.5	3.1.14
General	3.1.19
General	3.1.20
General	3.1.21
General	3.1.22
General	3.1.23
1.1	3.2.1
7.1, 7.2, 7.4, 7.7	3.2.2

6.6	3.2.3
6.4	3.2.4
1.1, 1.4	3.3.1.1
1.1, 1.4	3.3.1.2
1.1, 1.4	3.3.1.3
1.1, 1.4	3.3.1.4
1.5	3.3.1.5
1.5	3.3.1.6
1.2	3.3.1.7
1.2	3.3.1.8
1.3	3.3.1.9
1.6, 6.10.1	3.3.1.10
1.1	3.3.1.12
2.1.3, 7.2.1	3.3.2.1
2.1.1	3.3.2.2
2.1.1, 2.1.3	3.3.2.3
2.1.1	3.3.2.4
2.1.3	3.3.2.5
2.1.2	3.3.2.6
2.1.4	3.3.2.7
2.2.1, 2.2.2	3.3.2.8
2.3	3.3.2.9
2.3	3.3.2.10
2.4, 3.1, 3.3.4	3.3.2.11
3.1	3.3.2.12
1.1, 2.1.1, 2.1.3, 2.2.1, 2.2.2, 2.3, 3.6, 6.10.2, 7.1.3	3.3.2.13
3.3.1, 3.4.1, 3.5.3, 3.5.4	3.3.2.14
3.3.1	3.3.2.15
3.4.1	3.3.2.16
3.4.1	3.3.2.17

3.4.1	3.3.2.18
3.1, 3.4.2	3.3.2.19
3.5.3	3.3.2.20
3.5.4	3.3.2.21
3.4.2	3.3.2.22
3.3.4	3.3.2.23
3.5.1, 3.5.2, 4.7.5, 5.2.2, 5.3, 6.2	3.3.2.24
4.7.5, 3.5.1, 3.5.2, 5.3	3.3.2.25
2.1.1, 2.1.3, 2.2.1, 2.2.2, 2.3, 3.3.2, 3.3.3, 3.6, 6.10.2	3.3.2.26
2.1.1, 2.2.1, 2.2.2, 2.3, 3.3.3	3.3.2.27
3.2, 3.5.1	3.3.2.28
4.6.7, 4.6.8	3.3.2.29
5.2.1, 5.2.2	3.3.3.1
5.2.1, 5.2.2	3.3.3.3
5.2.1, 5.2.2	3.3.3.4
5.2.2	3.3.3.5
4.1.1, 4.1.2, 4.1.3, 4.1.4	3.3.4.1
4.7.5	3.3.4.2
4.1.5, 4.1.6	3.3.4.3
4.7.5	3.3.4.4
4.2	3.3.4.5
4.4, 4.5, 4.6.1, 4.6.2, 4.6.3, 4.6.4, 4.6.5, 4.6.6	3.3.4.7
5.4.1, 5.4.7	3.3.4.8
5.4.1, 5.4.2	3.3.4.9
5.4.2	3.3.4.10
6.2	3.3.4.11
1.1, 3.5.1, 3.5.2, 3.6, 4.2, 4.3, 4.6.7, 4.7.6, 5.2.2, 5.3, 6.2, 6.4	3.3.4.12
2.1.1, 2.2.2, 2.3, 3.6, 6.2	3.3.4.13
4.2, 4.14, 4.15, 4.16, 6.2	3.3.4.14
4.1.3, 4.2, 4.4, 4.5, 6.2	3.3.4.15

3.2, 4.4, 4.5, 4.6.1, 4.6.2, 4.6.3, 4.6.4, 4.6.5, 4.6.6, 4.6.7, 4.6.8, 5.3, 5.4.2, 6.2	3.3.4.16
7.1, 7.2	3.3.5.1
7.4, 7.7	3.3.5.2
7.3	3.3.5.3
7.5, 7.6	3.3.5.4
7.5, 6.1	3.3.5.5
6.1, 7.6	3.3.5.6
6.10.3, 7.8	3.3.5.7
6.1, 6.4, 6.5, 6.6, 6.9.1, 6.9.2, 6.9.3, 6.9.4, 6.9.5	3.3.6.1
6.10.2	3.3.6.2
6.10.2	3.3.6.3
6.1, 6.10.2	3.3.6.4
6.10.2	3.3.6.4.1
6.1, 6.10.3	3.3.6.5
6.10.3	3.3.6.5.1
6.5	3.3.6.6
2.2.1, 2.2.2, 2.3, 2.1.1, 3.3.3, 6.3, 4.1.7, 4.1.8	3.3.6.7
6.7	3.3.6.8
6.5	3.3.6.9
5.5	3.3.6.10

---

## Acronyms

ACCA	Automatic Cloud Cover Assessment
ADS	angular display sensor
AOS	Advanced Orbiting Systems
ASCII	American Standard Code for Information Interchange
BCH	Bose-Chaudhuri-Hocquenghem
BER	bit error rate
CADU	channel access data unit
CASE	computer-aided software engineering
CCB	Configuration Control Board
CIS	Centralized Information System
COTS	commercial off-the-shelf
CCSDS	Consultative Committee for Space Data Systems
CLCW	command link control word
CNMOS	Consolidated Network and Mission Operations Support
CRC	cyclic redundancy check
CRUD	create, retrieve, update, and delete
CSC	Computer Sciences Corporation
DAA	data availability acknowledgment
DAMT	Distributed Application Monitor Tool
DAN	data availability notice
DBMS	database management system
DDF	Data Distribution Facility
DDL	Data Definition Language
DDN	data delivery notice
DFCB	Data Format Control Book
DFD	data flow diagram
DPCP	Distributed Process Control Program
DSN	Deep Space Network
EDC	EROS Data Center
EOL	end of line
ER	entity relationship
ERD	entity relationship diagram
EROS	Earth Resources Observation System
ETM+	Enhanced Thematic Mapper Plus
FHS ERR	first half scan error
F&PS	functional and performance specification

GCI	geocentric inertial
GOTS	Government off-the-shelf
GSFC	Goddard Space Flight Center
GUI	graphical user interface
IAS	Image Assessment System
ICD	interface control document
IDPS	Image Data Processing Subsystem
IM	Information Modeling
IPC	interprocess communication
LDTs	LPS data transfer subsystem
LGS	Landsat Ground Station
LPS	Landsat 7 Processing System
LP DAAC	Land Processes Distributed Active Archive Center
LZP	level-zero processing
L0R	level-zero reformatted
MACS	management and control subsystem
MFPS	major frame processing subsystem
MMAS	Martin Marietta Astro Space
MSCD	mirror scan correction data
MOC	Mission Operations Center
MO&DSD	Mission Operations and Data Systems Directorate
NASA	National Aeronautics and Space Administration
NCC	Network Control Center
NOAA	National Oceanic and Atmospheric Administration
Pacor	Packet Processing
PCD	payload correction data
PCDS	PCD processing subsystem
PRN	pseudorandom noise
RDCS	raw data capture subsystem
RDPS	raw data processing subsystem
RMA	reliability, maintainability, and availability
RS	Reed-Solomon
SCLF	search/check/lock/flywheel
SCN DIR	scan direction
SDL	Storage Definition Language

SDPF	Spacelab Data Processing Facility
SDS	system design specification
SGI	Silicon Graphics, Inc.
SHS ERR	second half scan error
SQL	Structured Query Language
SRS	software requirements specification
SWCI	software configuration item
UIL	User Interface Language
USGS	United States Geological Survey
UTC	universal time coordinated
VCDU	virtual channel data unit
VCID	virtual channel identifier
WRS	World wide Reference System